

FAKTORZEHN

Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

04 Customizing & Tools



Amir Aboueldahab

Senior Developer

- Seit 2018 bei Faktor Zehn GmbH
- Verschiedene kleinere Projekteinsätze
- Seit 2020 im Kundenprojekt bei der Union Reiseversicherung

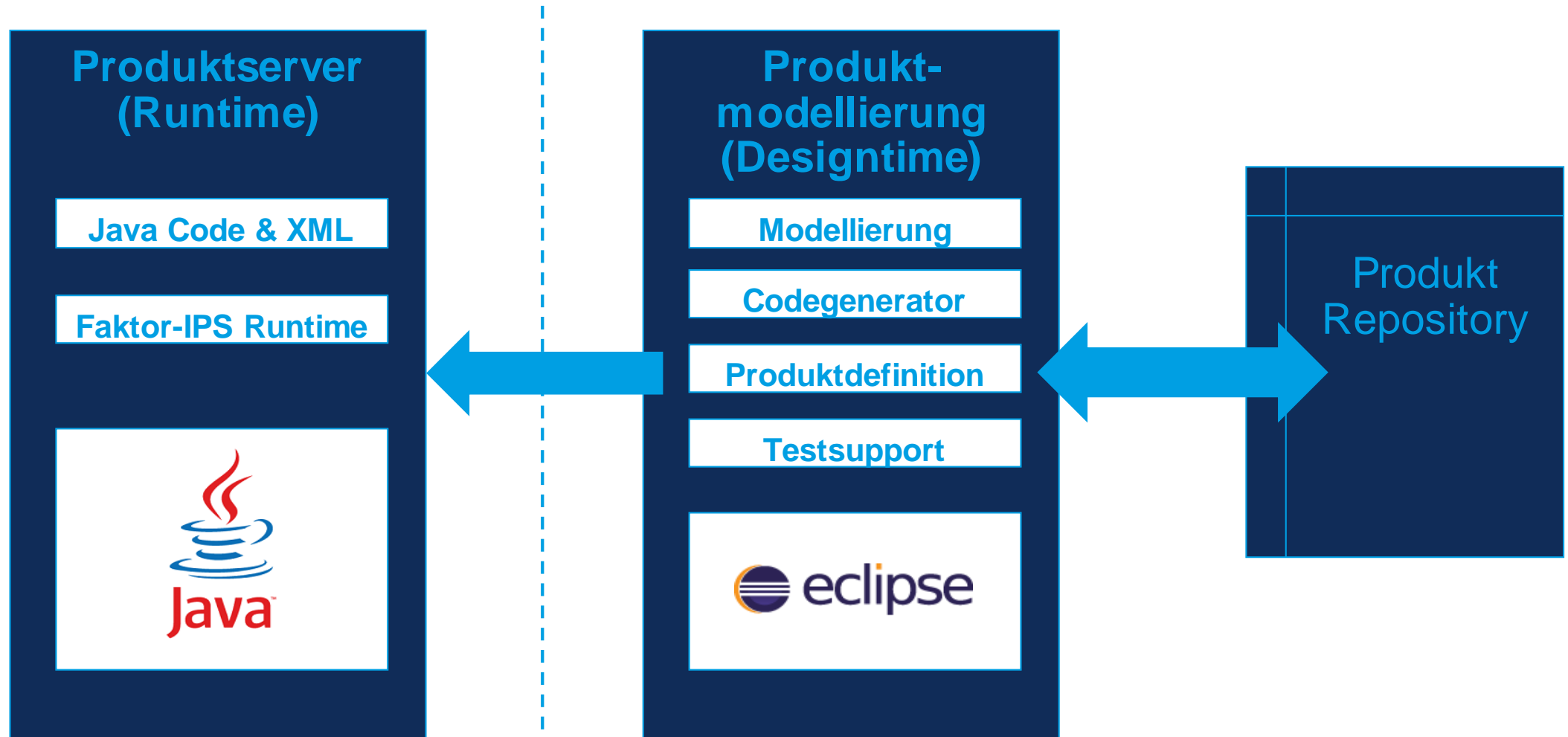


Niels Kammerer

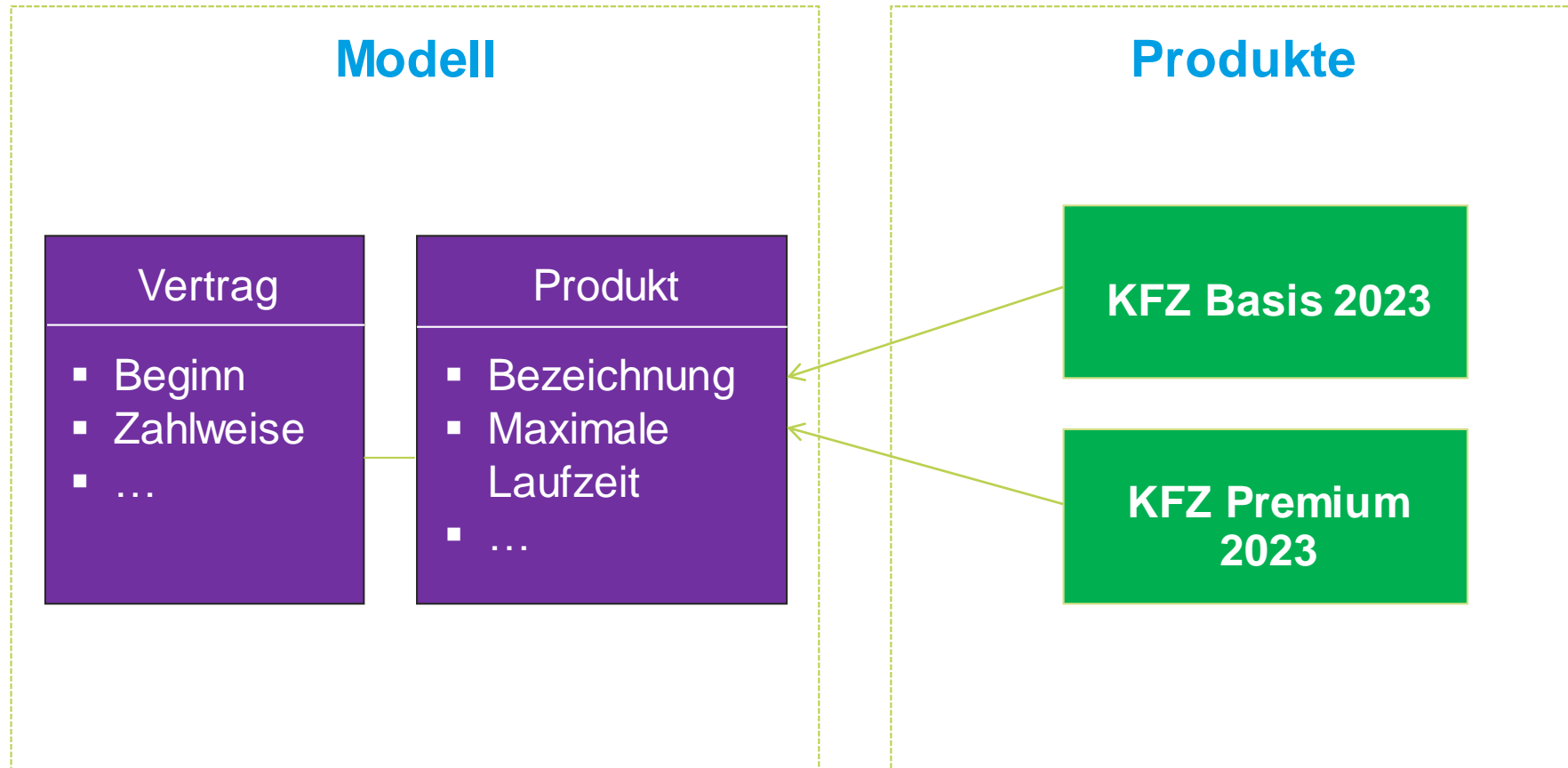
Senior Developer

- Seit 2019 bei Faktor Zehn GmbH
- Zu Beginn in der Produktentwicklung
- Seit 2021 im Kundenprojekt HDI Cyber

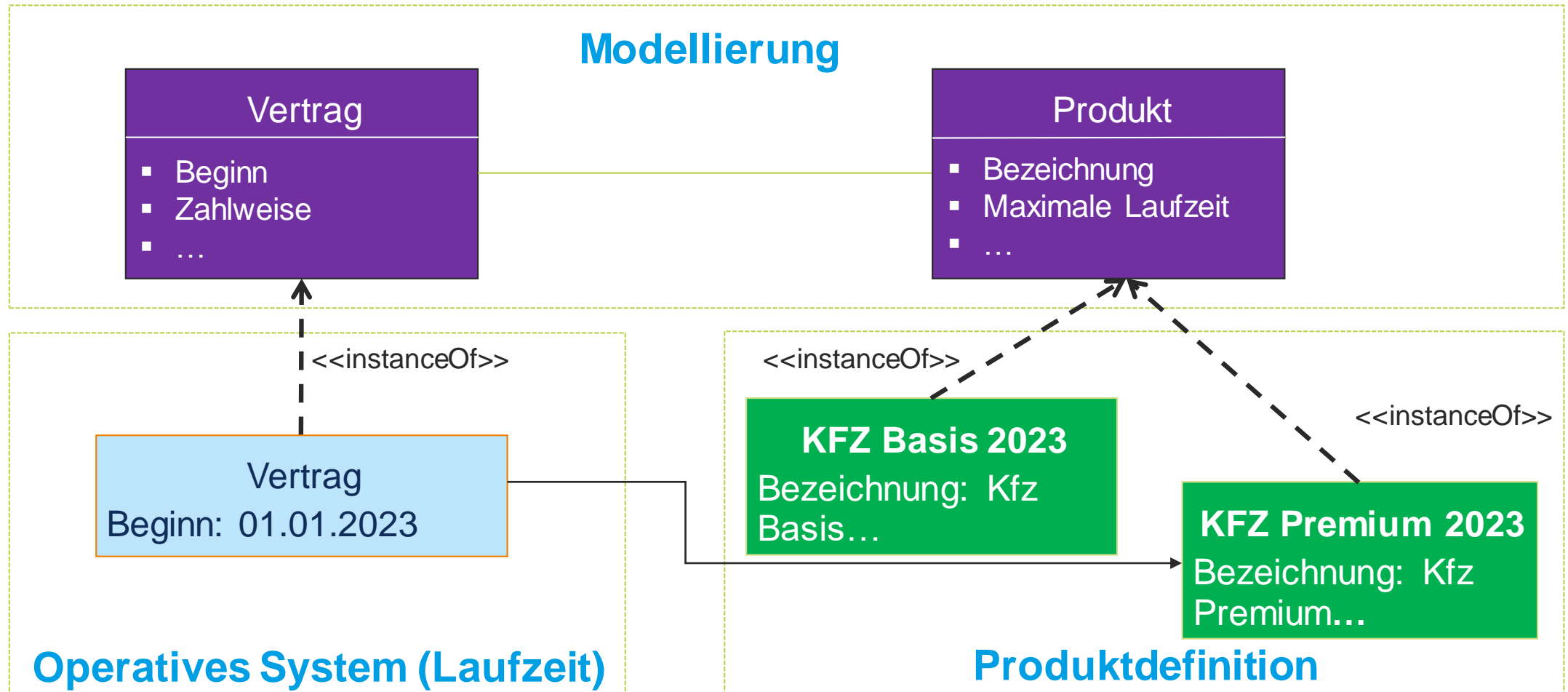
Faktor-IPS ist ein Modellierungs- und Produktdefinitions- werkzeug auf Basis von Java & Eclipse.



Konkrete Produkte basieren auf einem Vertrags- und Produktmodell



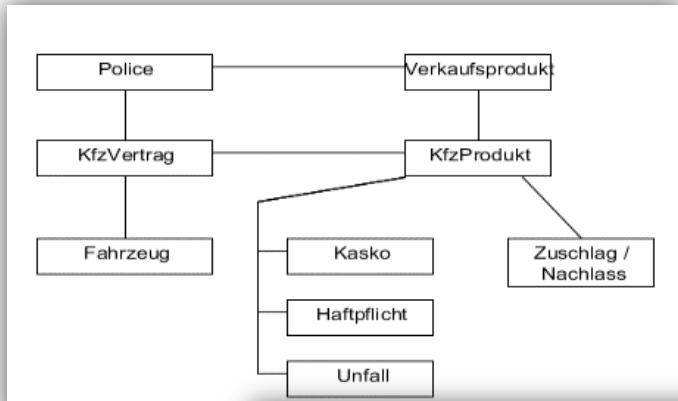
Operative Systeme arbeiten mit Instanzen des Vertragsmodells und nutzen die definierten Produkte



Faktor-IPS ist ein Werkzeug für Produktentwickler und Anwendungsentwickler

Modell & Java Code

Anwendungsentwicklung



```
@Override
public Long getKmStand() {
    return kmStand;
}

@Override
public void setKmStand(Long newValue) {
    Long oldKmStand = kmStand;
    this.kmStand = newValue;
    notifyChangeListeners(new PropertyChangeEvent(this,
        PROPERTY_KMSTAND, oldKmStand, kmStand));
}

@Override
public Integer getTypklasseHaftpflicht() {
```

Produkte

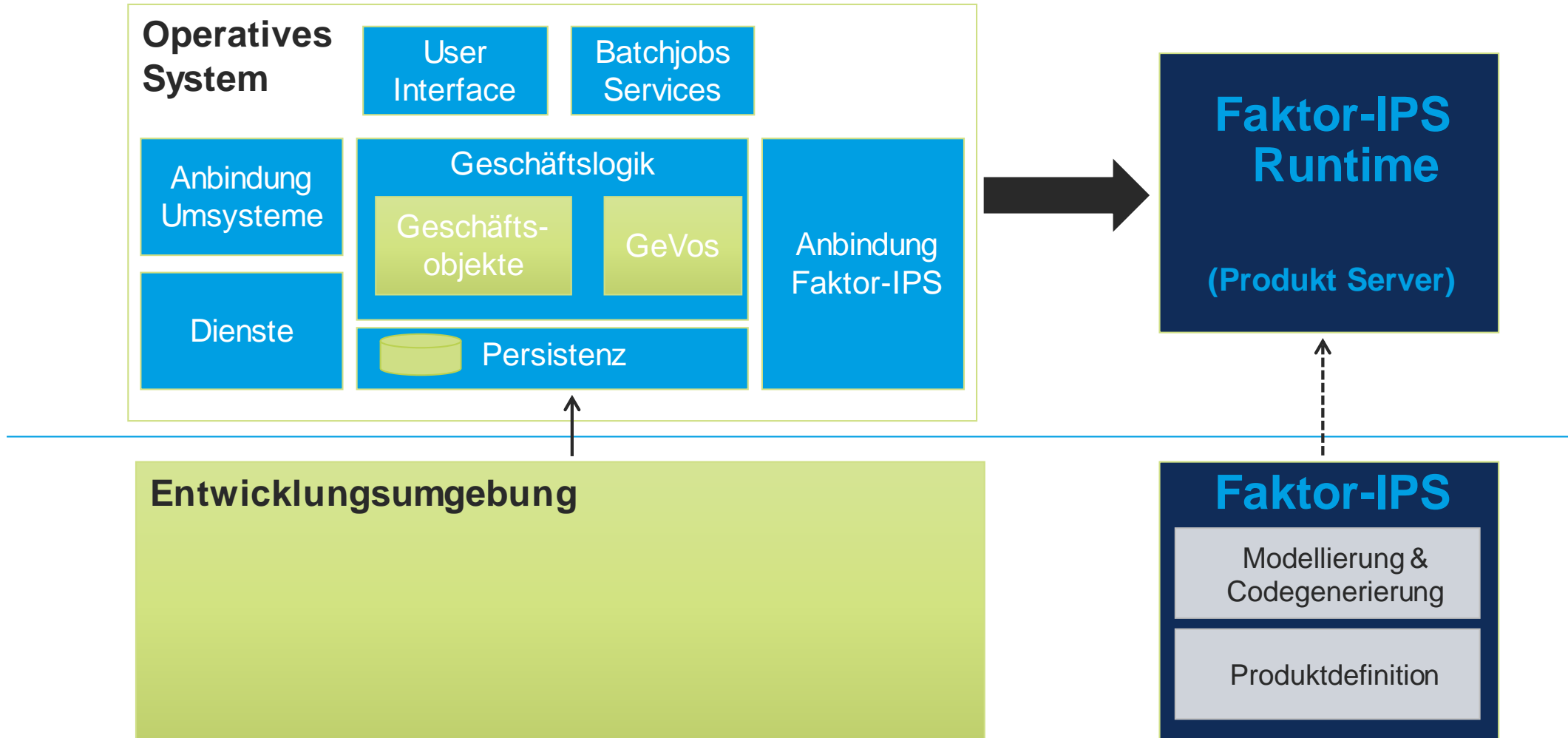
Produktentwicklung

Produktbausteine

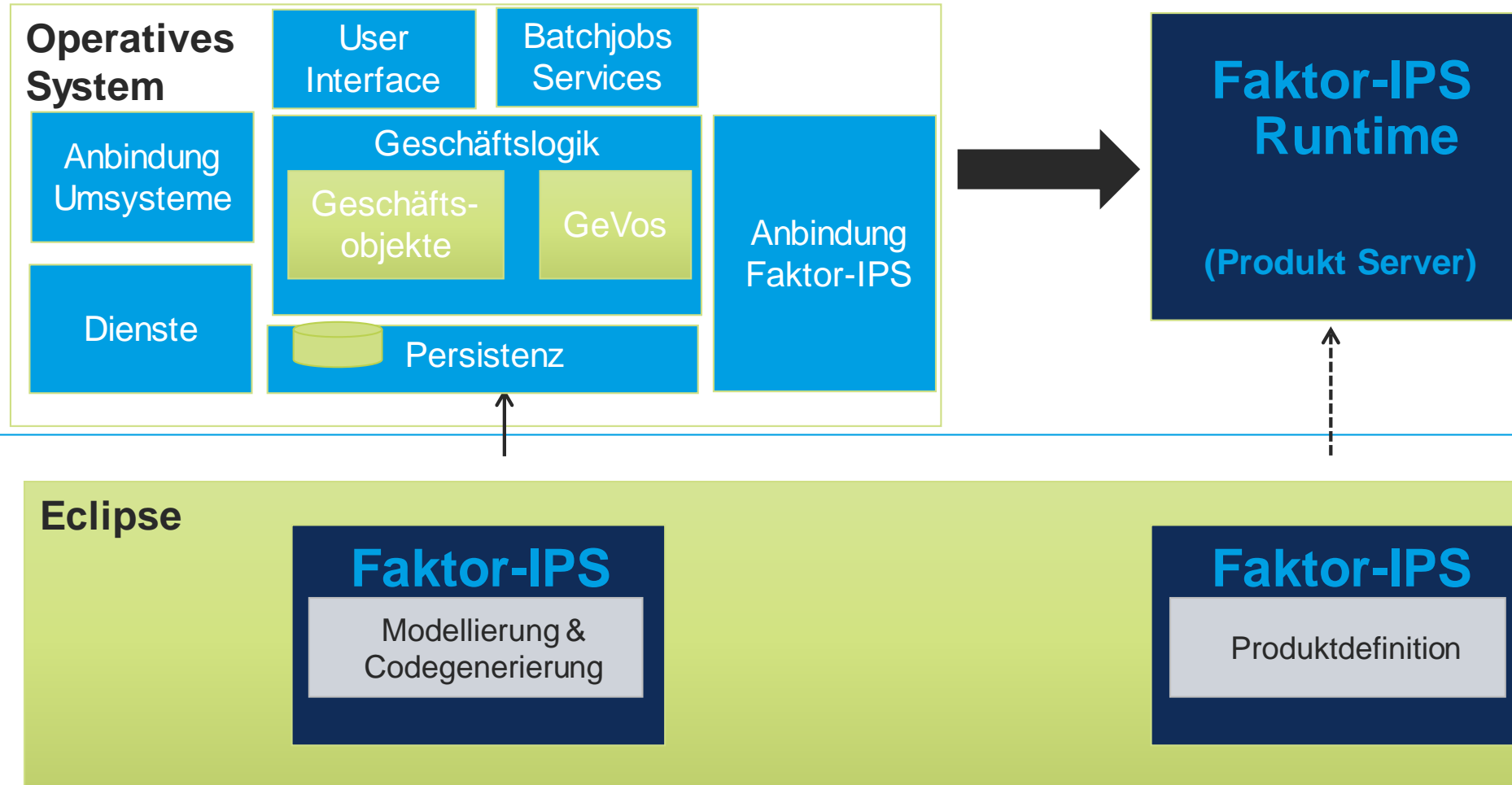
Tabellen

Formeln

Faktor-IPS kann als klassisches Produktsystem eingesetzt werden.



Faktor-IPS kann als Entwicklungswerkzeug für ein operatives System und als Produktkonfigurator verwendet werden.



Ziel der Übungen

- Mini-Tarifrechner für Hausrat
- Einführung einer neuen Produktgeneration

Agenda

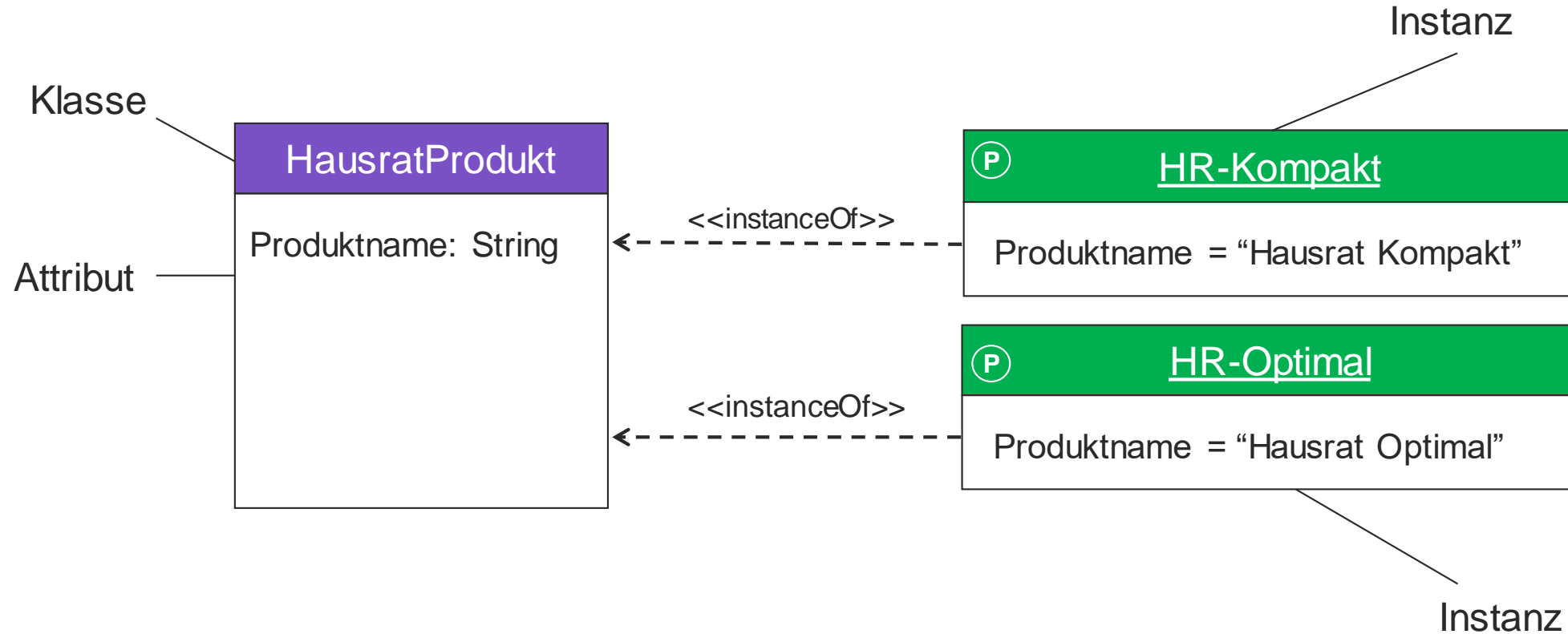
01 Motivation

02 UML Refresh

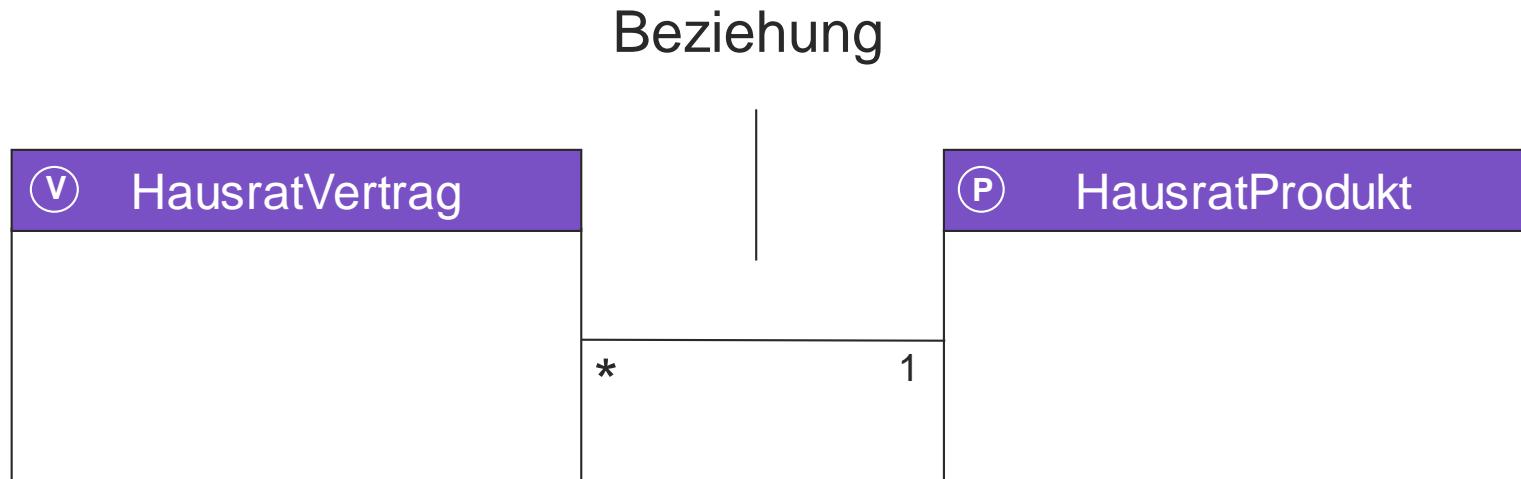
03 Modellierung & Produktdefinition mit Faktor-IPS

04 Customizing & Tools

Klassen, Attribute und Instanzen

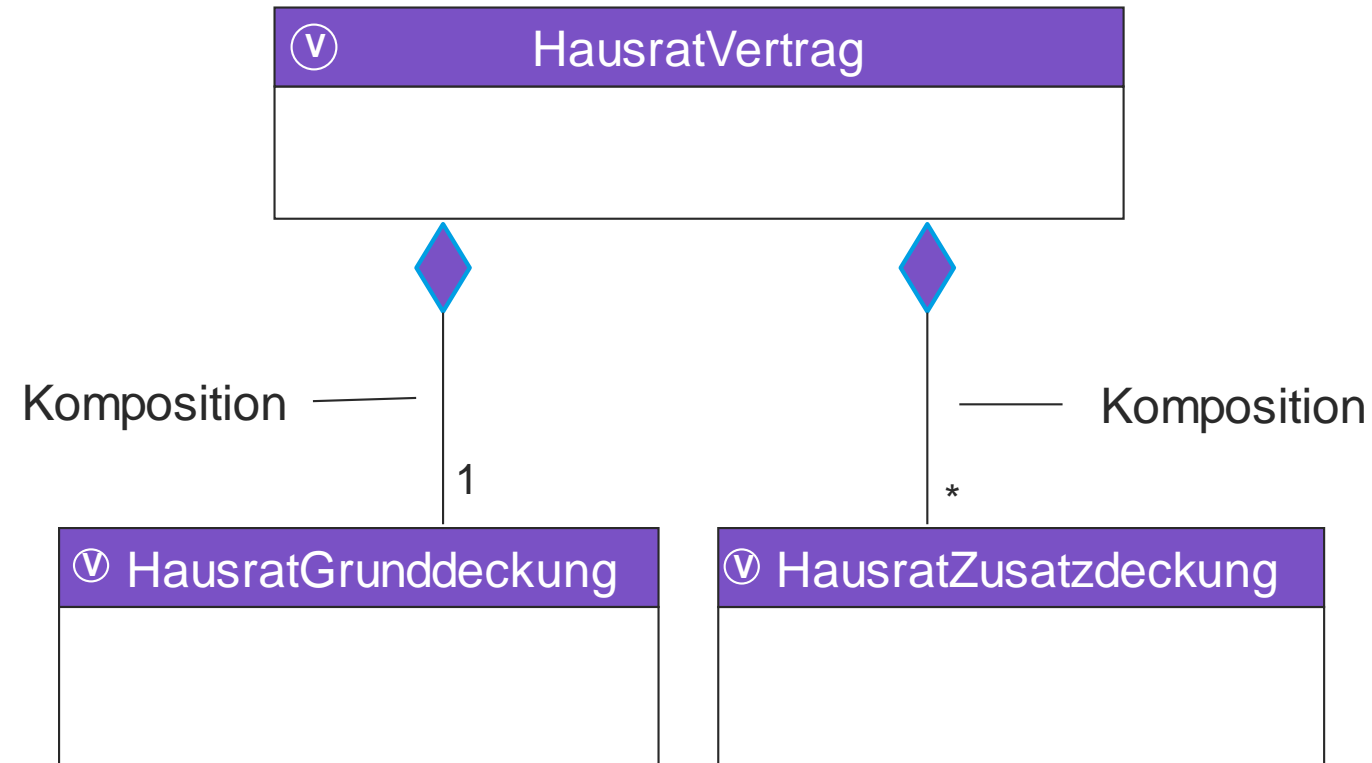


Beziehungen



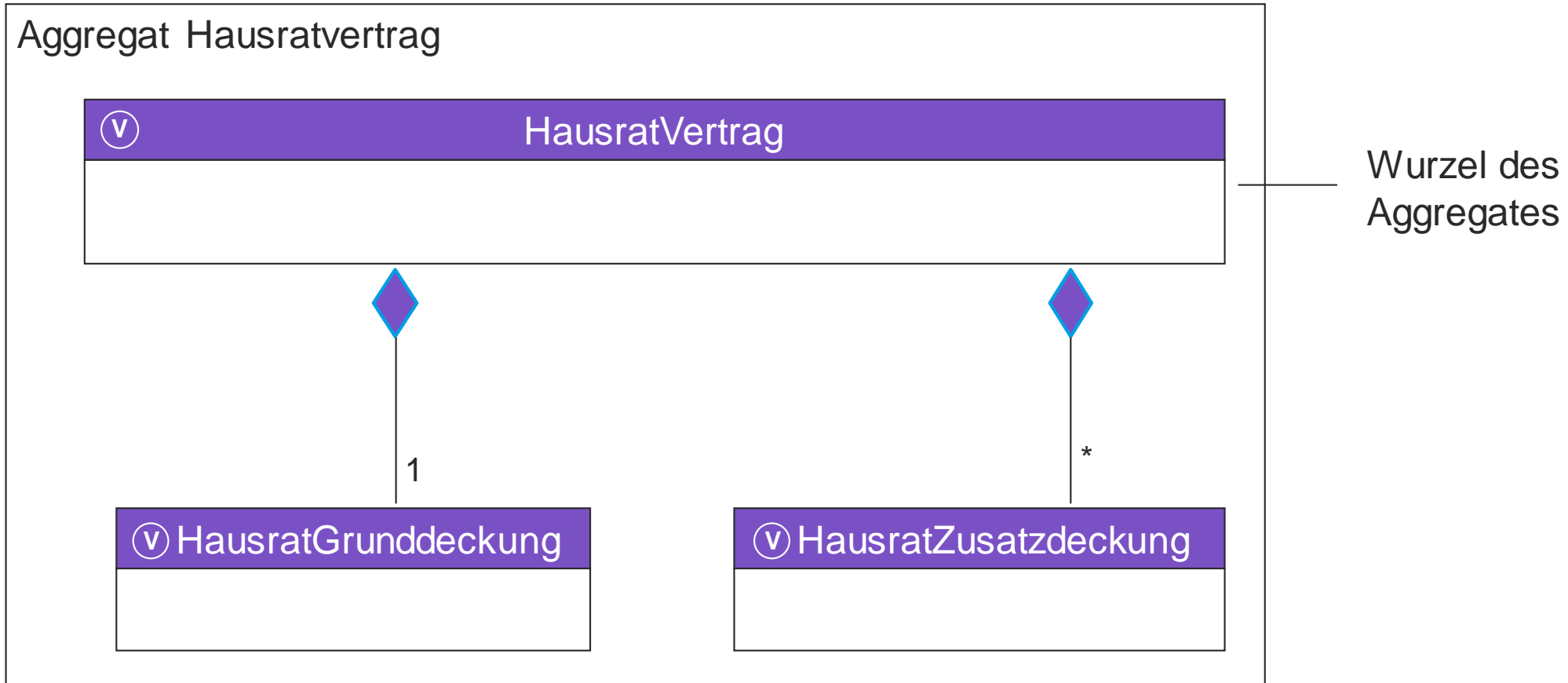
Jeder Hausratvertrag basiert auf genau einem Hausratprodukt.
Auf Basis eines Hausratproduktes können beliebig viele Hausratverträge abgeschlossen werden.

Kompositionen

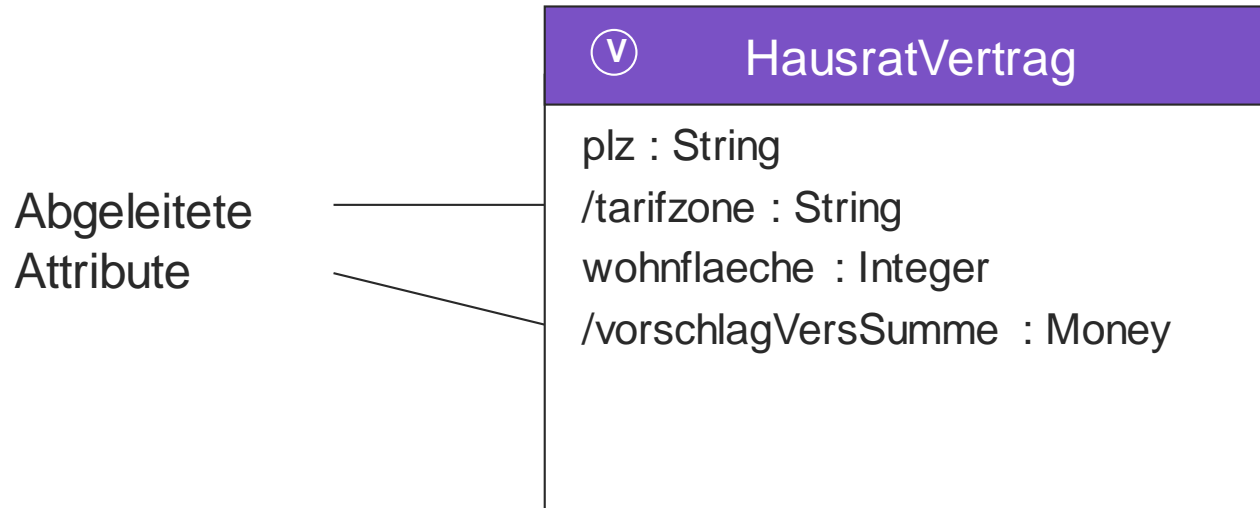


Hausratgrunddeckung und Hausratzusatzdeckungen sind Bestandteile des Hausratvertrags.

Aggregat & Wurzel des Aggregates



Abgeleitete Attribute (Derived)



Tarifzone wird aus der Postleitzahl ermittelt.

Vorschlag für die Versicherungssumme wird aus der Wohnfläche ermittelt.

Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Projekt anlegen

- Maven-Projekt „Hausratmodell“ per Maven Archetype erstellen
- Projekteinstellungen: Alle unnötigen Generator-Optionen deaktivieren

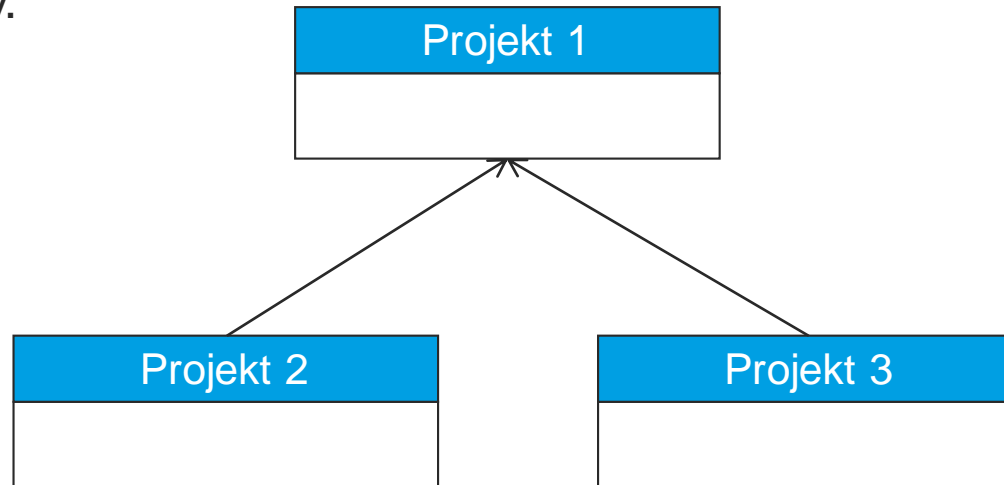
Übungen zu Kapitel III.A

- analog zur Demo
- Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



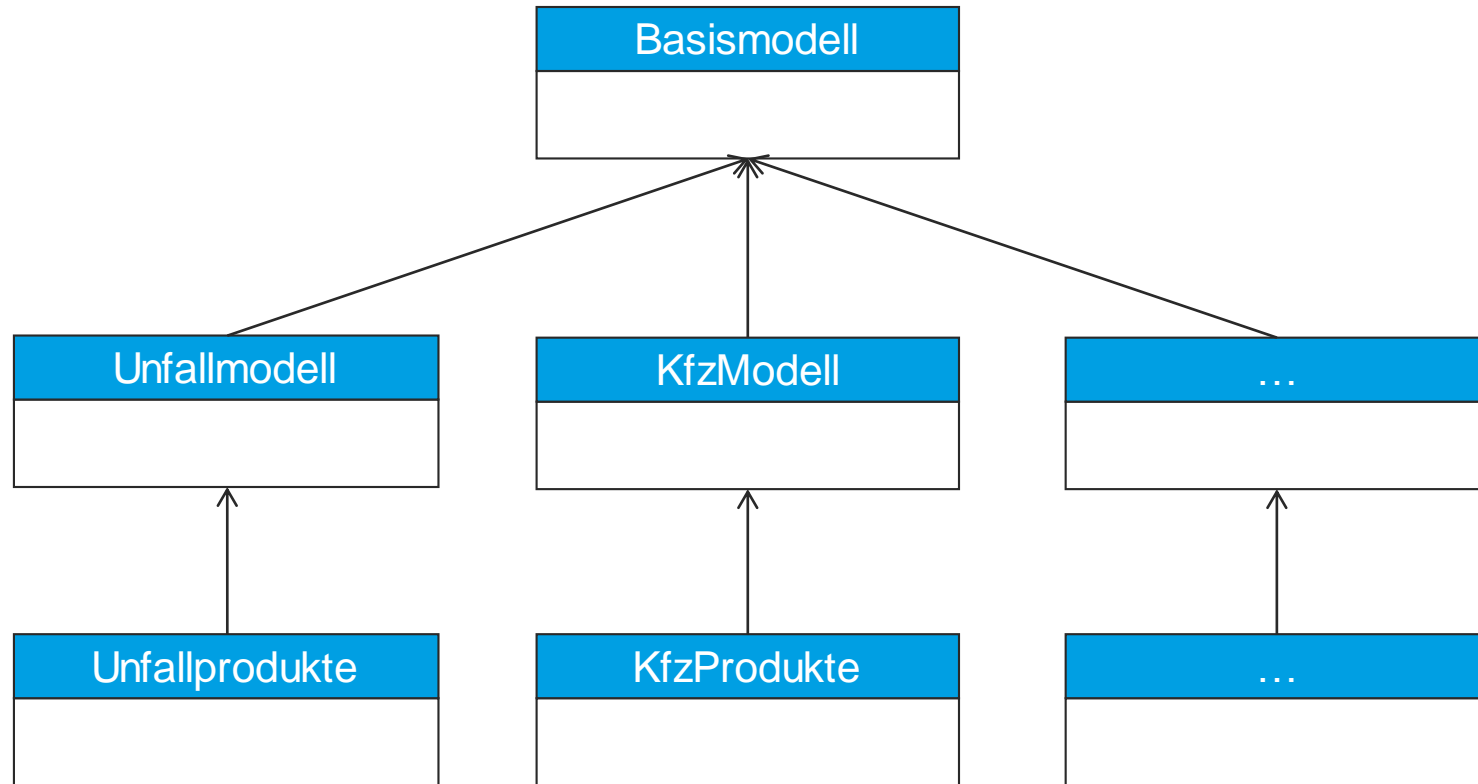
Umgang mit mehreren Projekten

- Faktor-IPS kann mit beliebig vielen Projekten umgehen.
- Ein Projekt kann andere Projekte referenzieren. Damit können alle Objekte aus den referenzierten Projekten verwendet werden.
- Die Beziehung ist transitiv.

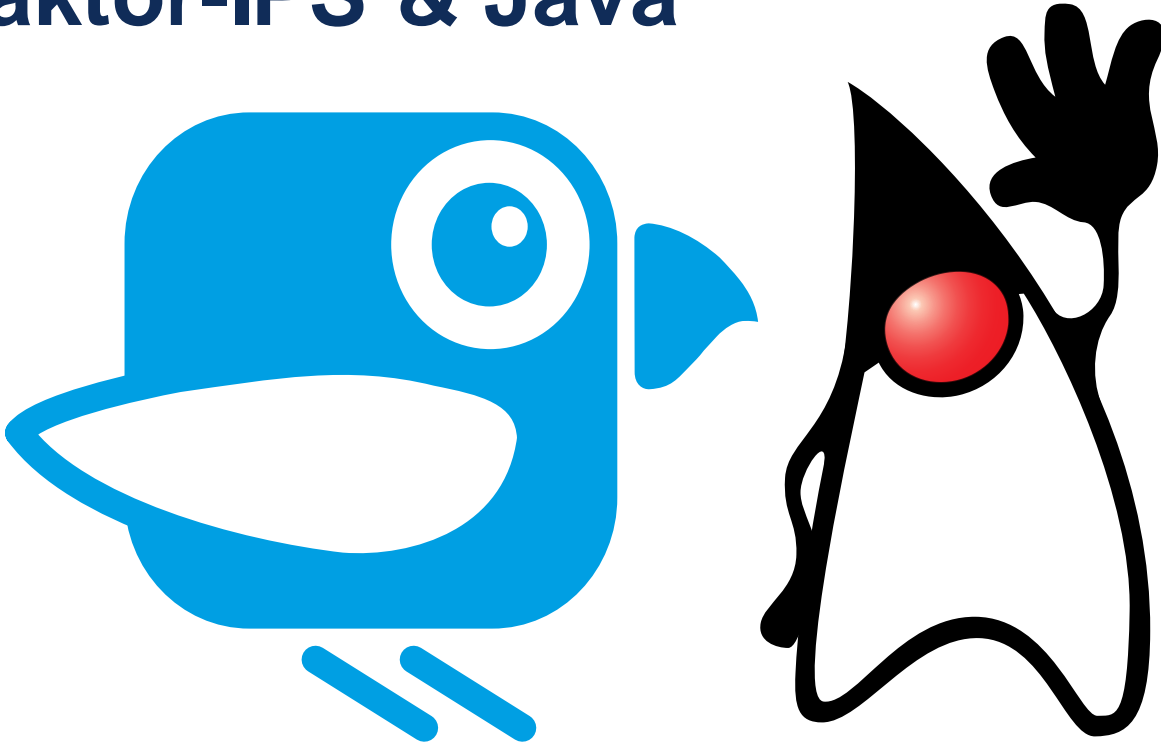


In Projekt 2 sind alle Objekte aus dem Projekt 2 selbst und alle Objekte aus Projekt 1 sichtbar. Objekte aus Projekt 3 sind nicht sichtbar.

Übliche Projektstruktur



Faktor-IPS & Java



- Abhängigkeiten müssen parallel für Java **und** Faktor-IPS konfiguriert werden
- Meist weitere Java-Abhängigkeiten
- Beide können Maven nutzen
 - Siehe <https://www.faktorzehn.org/de/dokumentation/verwendung-von-faktor-ips-projekten-als-maven-dependencies/>

Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

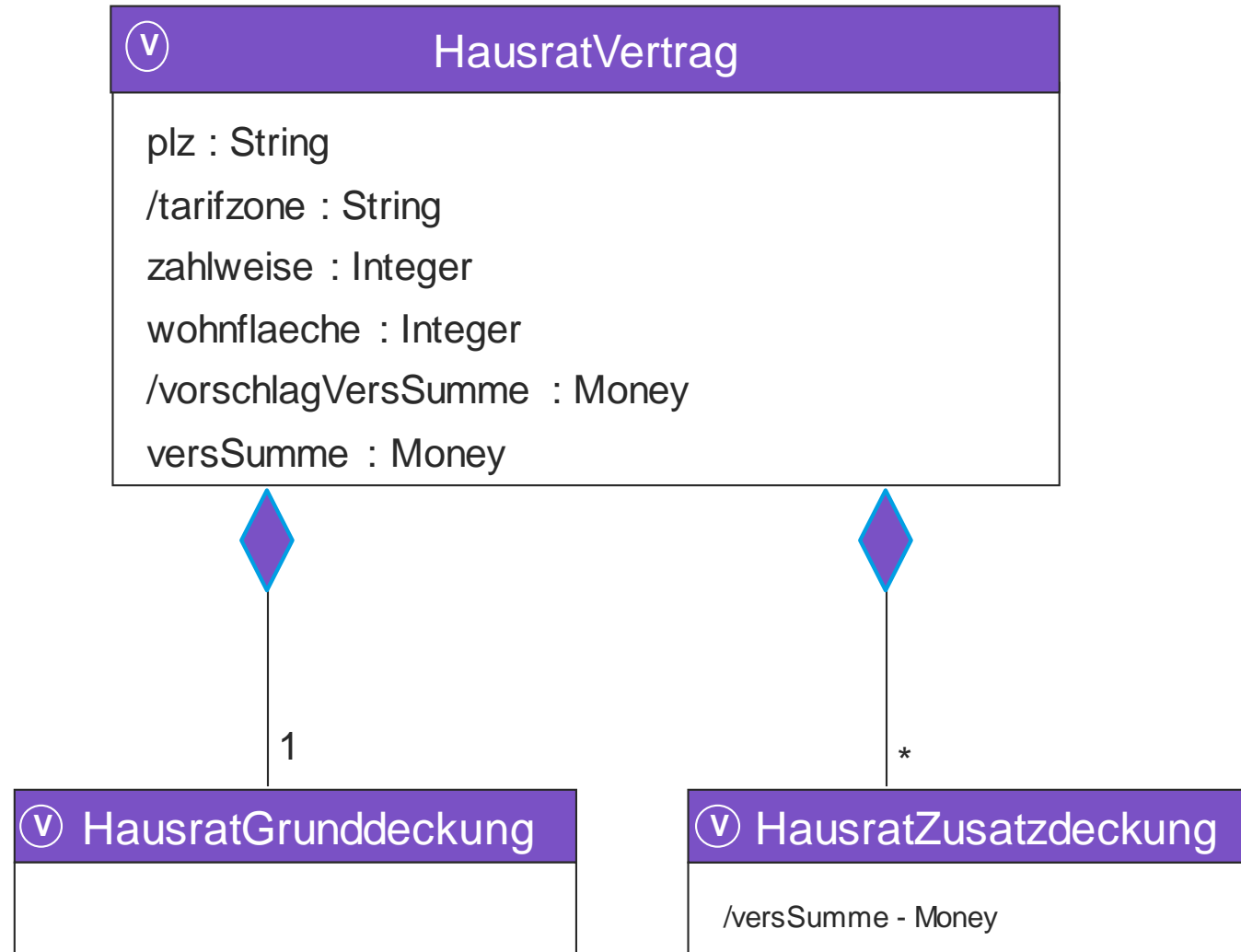
Modellierung der Vertragsseite

Attribute

Beziehungen

04 Customizing & Tools

Schulungsbeispiel: Hausratmodell (ohne Produktinformationen)



Demo: Erste Klasse anlegen

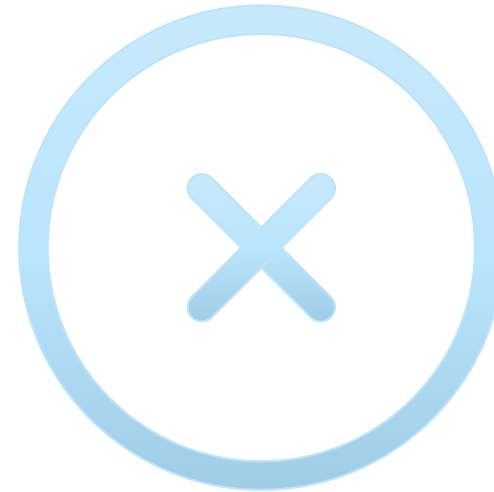
v

HausratVertrag

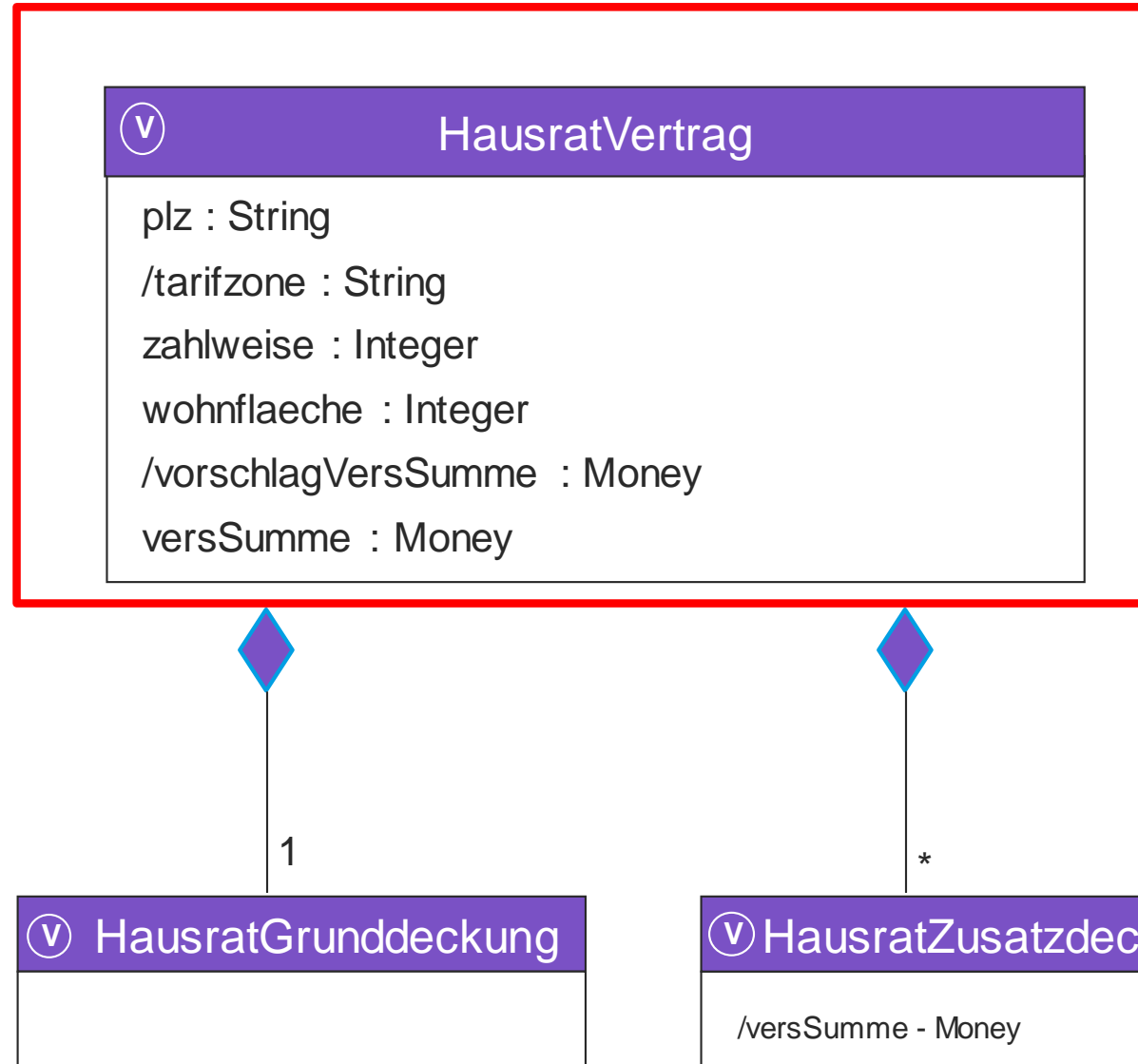
- IPS-Package „hausrat“ anlegen
- Vertragsteil-Typ „HausratVertrag“ anlegen
- Generierten Sourcecode erläutern
- Faktor-IPS Modellexplorer erläutern

Übungen zu Kapitel III.B

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Modellierung von Attributen



Demo

Folgende Attribute anlegen:

- plz
- zahlweise
- tarifzone

Umgang mit generiertem Code demonstrieren.

Umgang mit generiertem Code

- Der von Faktor-IPS generierte Javacode kann vom Entwickler geändert werden.
- Damit der Code-Generator vom Entwickler angepasstem Code nicht überschreibt sind verschiedene Einstellungen möglich. Diese werden über sogenannte „Java Doc Custom Tags“ gesteuert.
- An Klassen/Methoden/Feldern.

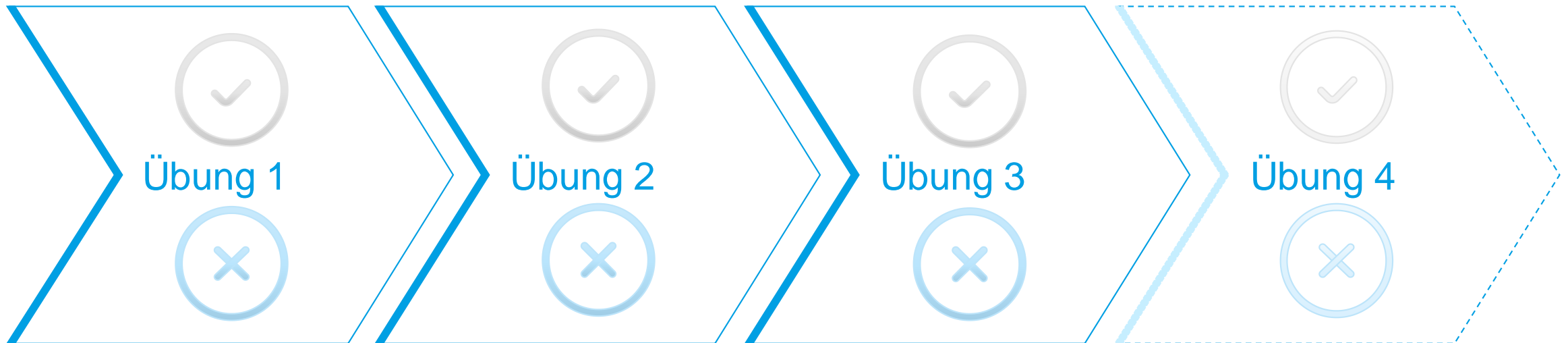
| Schlüsselwörter | Erläuterung |
|-----------------------|---|
| @generated | Der folgende Code wird vom Generator überschrieben |
| @generated NOT | Der folgende Code wird nicht vom Generator überschrieben |
| @restrainedmodifiable | Teile des Codes einer Methode können vom Entwickler angepasst werden. |

Weitere Optionen

| Schlüsselwörter | Erläuterung |
|--|---|
| @implements <Interface> | Lässt die generierte Klasse zusätzlich das angegebene Interface implementieren (für generierte Interfaces kann auch @extends verwendet werden) |
| @customizedAnnotations [ALL] | Die Annotationen werden nicht vom Generator überschrieben. Der Zusatz ‚ALL‘ kann weggelassen werden. |
| @customizedAnnotations ADDED | Der Entwickler hat Annotationen hinzugefügt. Der Generator darf daher keine Annotationen mehr löschen. |
| @customizedAnnotations REMOVED | Der Entwickler hat generierte Annotationen entfernt. Der Generator darf daher keine neuen Annotationen hinzufügen |
| @customizedAnnotations CONTENT-CHANGED | Der Entwickler hat die Attribute einer Annotation verändert. Der Generator darf den Inhalt nicht mehr überschreiben. |

Übungen zu Kapitel III.B.1

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

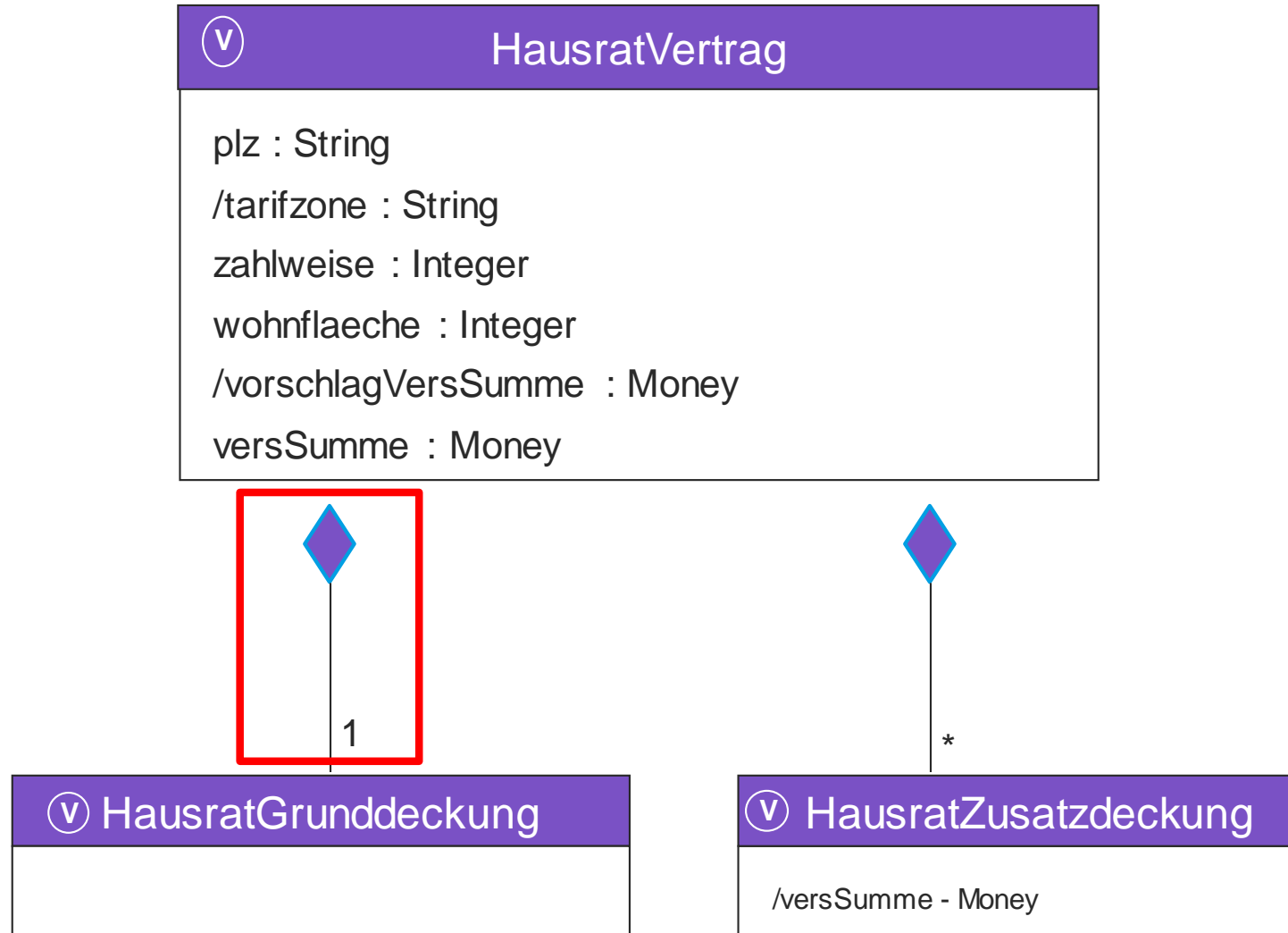
Modellierung der Vertragsseite

Attribute

Beziehungen

04 Customizing & Tools

Modellierung von Beziehungen



Demo

- Anlegen der Klasse HausratGrunddeckung
- Anlegen der Beziehung zwischen HausratVertrag und HausratGrunddeckung.
- Analyse des generierten Sourcecodes.

Übungen zu Kapitel III.B.2

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Agenda

01

Motivation

02

UML Refresh

03

Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Grundlagen & Produktattribute

Beziehungen

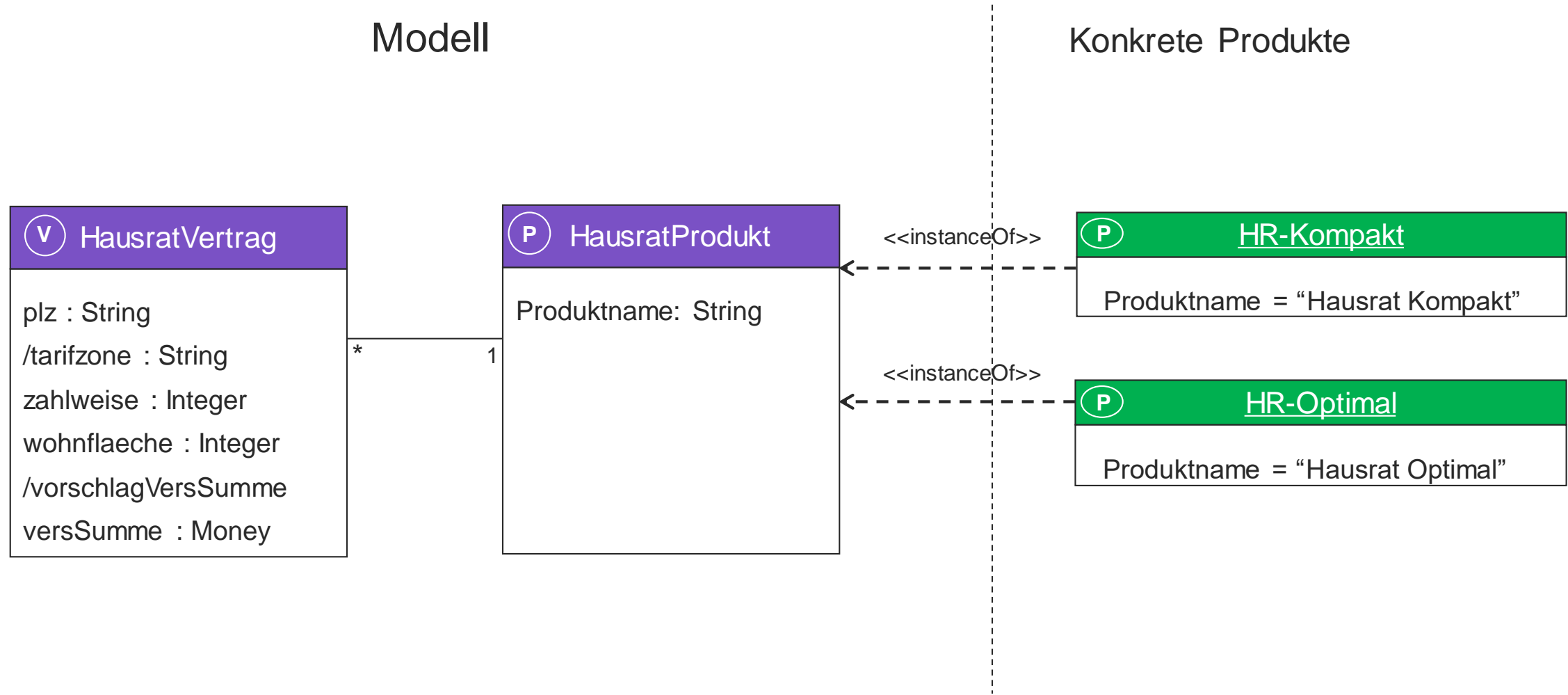
Konfigurierbare Vertragsattribute

Zugriff auf Produktdaten zur Laufzeit

Motivation

- Es gibt zwei Hausratprodukte:
 - Hausrat Kompakt: Günstiger Basisschutz
 - Hausrat Optimal: Optimaler Schutz
-
- Jeder Hausratvertrag wird entweder auf Basis von Hausrat Kompakt oder Hausrat Optimal abgeschlossen.

Abbildung der Hausratprodukte

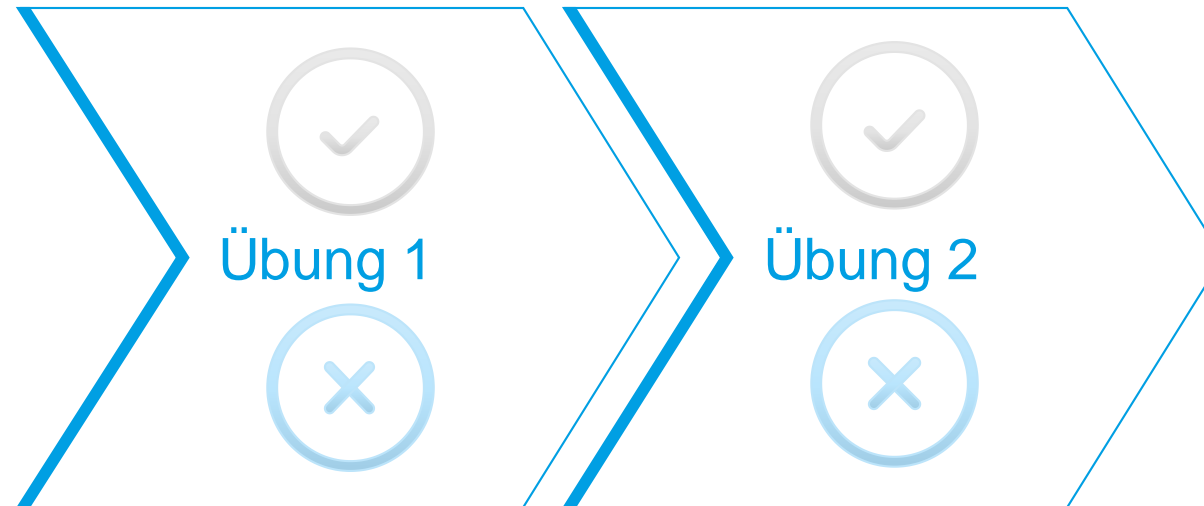


Demo

- Produktbaustein-Typ „HausratProdukt“ anlegen
- Attribut „produktname“ definieren. (Checkbox „Änderungen im Zeitablauf“ keinen Haken)
- In die Produktdefinitionsperspektive wechseln
- Produkte HR-Kompakt & HR-Optimal anlegen
- In die Java-Perspektive wechseln
- Attribut „kurzbezeichnung“ anlegen.
- In die Produktdefinitionsperspektive wechseln
- Kurzbezeichnung in den beiden Produkten pflegen.

Übungen zu Kapitel III.C.1

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

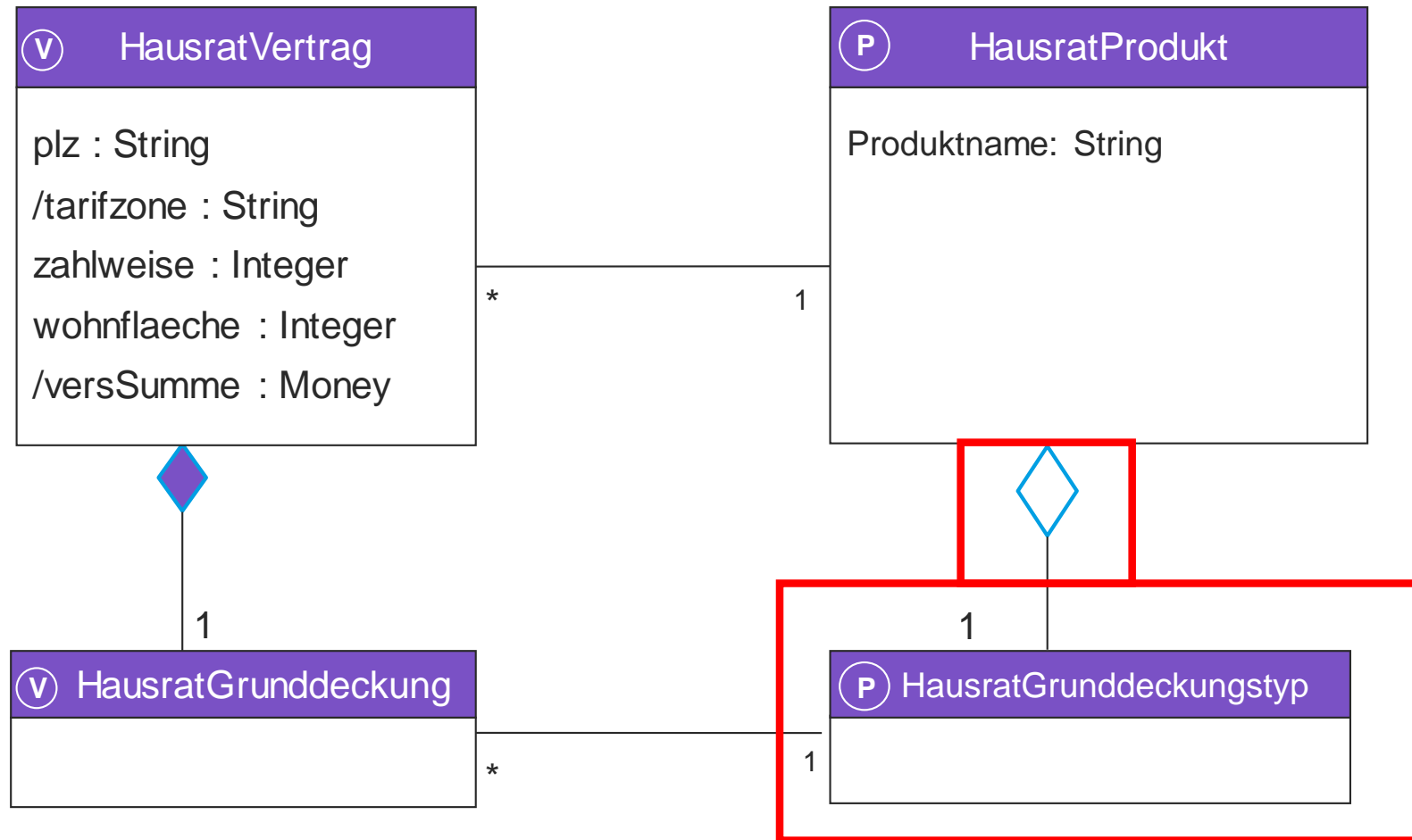
Grundlagen & Produktattribute

Beziehungen

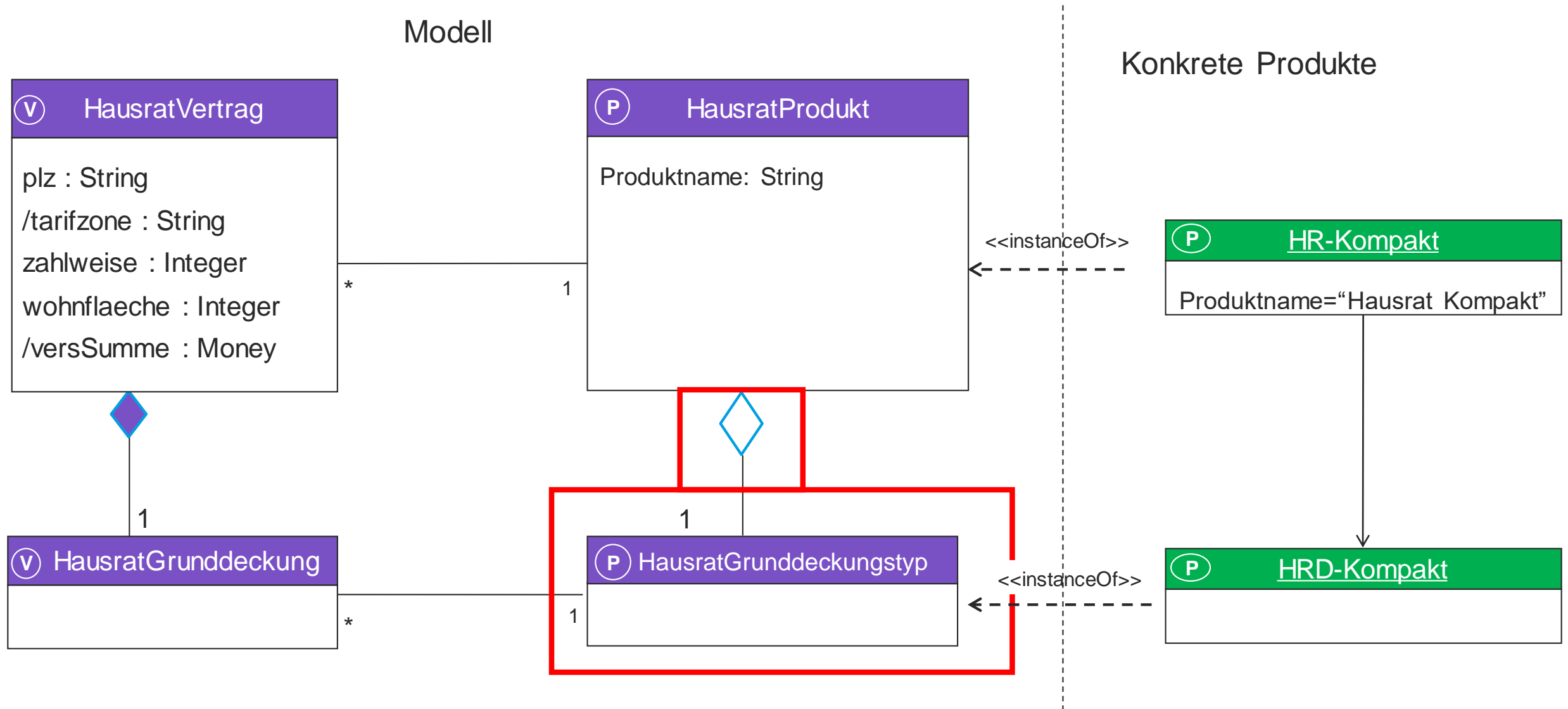
Konfigurierbare Vertragsattribute

Zugriff auf Produktdaten zur Laufzeit

Beziehungen auf Produktseite



Beziehungen auf Produktseite (inkl. Produktbausteine)

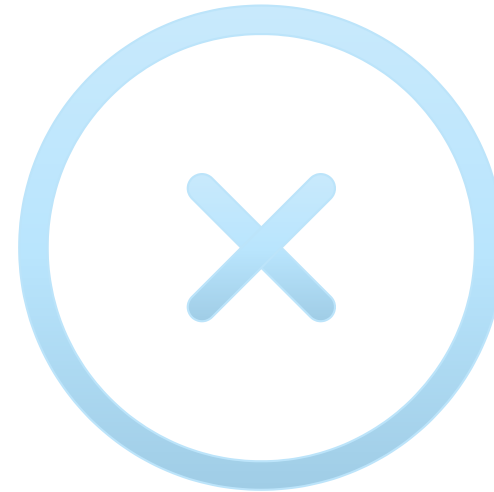


Demo

- Produktbausteintyp Hausratgrunddeckungstyp anlegen und mit HausratGrunddeckung verknüpfen.
- Beziehung zwischen Hausratprodukt und HausratGrunddeckungstyp anlegen.
- Darauf achten, dass die Beziehung nicht änderbar ist im Zeitablauf.
- Wechseln in die Produktdefinitionsperspektive.
- Grunddeckung für Hausrat Kompakt anlegen und Beziehung per Drag & Drop herstellen.
- Hausrat Optimal im Produktstruktur-Explorer öffnen
- Grunddeckung für Hausrat Optimal per “Add new...” anlegen.

Übungen zu Kapitel III.C.2

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Grundlagen & Produktattribute

Beziehungen

Konfigurierbare Vertragsattribute

Zugriff auf Produktdaten zur Laufzeit

Konfigurationsmöglichkeit eines Hausratvertrages

| Änderbare Eigenschaft des Hausratvertrags | Konfigurationsmöglichkeiten im Produkt |
|---|--|
| zahlweise | Die im Vertrag erlaubten Zahlungsweisen. Der Vorbelegungswert für die Zahlungsweise bei Erzeugung eines neuen Vertrags. |
| wohnflaeche | Bereich (min, max), in dem die Wohnfläche liegen muss. Der Vorbelegungswert für die Wohnfläche bei Erzeugung eines neuen Vertrags. |
| versSumme | Bereich, in dem die Versicherungssumme liegen muss. Der Vorbelegungswert für die Versicherungssumme bei Erzeugung eines neuen Vertrags. |

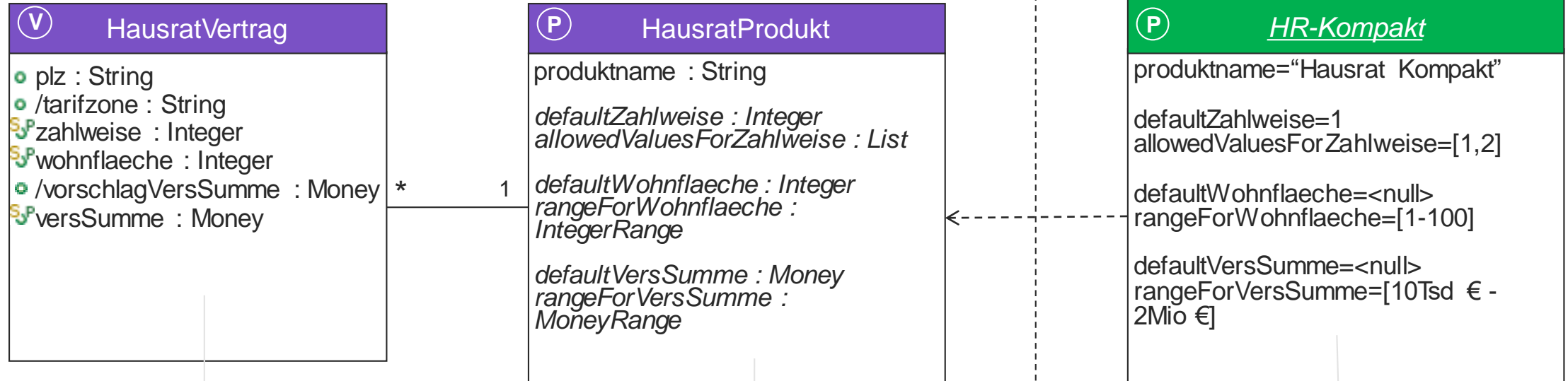
Beispielprodukte: HR-Kompakt & HR-Optimal

| Konfigurationsmöglichkeit | HR-Kompakt | HR-Optimal |
|------------------------------|------------------------|--|
| Vorbelegungswert Zahlweise | jährlich | jährlich |
| Erlaubte Zahlweisen | halbjährlich, jährlich | monatlich, vierteljährlich, halbjährlich, jährlich |
| Vorbelegungswert Wohnfläche | <null> | <null> |
| Erlaubter Bereich Wohnfläche | 0-100 qm | 0-200 qm |
| Defaultwert VersSumme | <null> | <null> |
| Erlaubter Bereich VersSumme | 10Tsd – 2Mio Euro | 10Tsd – 5Mio Euro |

Konfigurierbare Vertragsattribute – Modell & Bausteine

Modell

Konkrete Produkte



Änderbare Attribute werden als konfigurierbar gekennzeichnet.

Implizit gibt es im Produktbaustein-Typ Attribute für den Vorbelegungswert und die erlaubten Werte. Diese brauchen nicht explizit im Modell erfasst werden!

In den Bausteinen können der Defaultwert und die erlaubten Werte definiert werden

Demo: Modellerweiterung in Faktor-IPS

- Markieren des Attributes „zahlweise“ als konfigurierbar und „Typ der Wertemenge“ auf „Aufzählung“ festlegen
- Angabe der möglichen Zahlweisen in beiden Produkten
- Markieren des Attributes „wohnflaeche“ als konfigurierbar und „Typ der Wertemenge“ auf „Bereich“ festlegen
- Angabe des erlaubten Bereichs für die Wohnfläche in den beiden Produkten
- Markieren des Attributes „versSumme“ als konfigurierbar und „Type der Wertemenge“ auf „Bereich“ festlegen
- Angabe des erlaubten Bereichs für die Versicherungssumme

Übungen zu Kapitel III.C.3

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Grundlagen & Produktattribute

Beziehungen

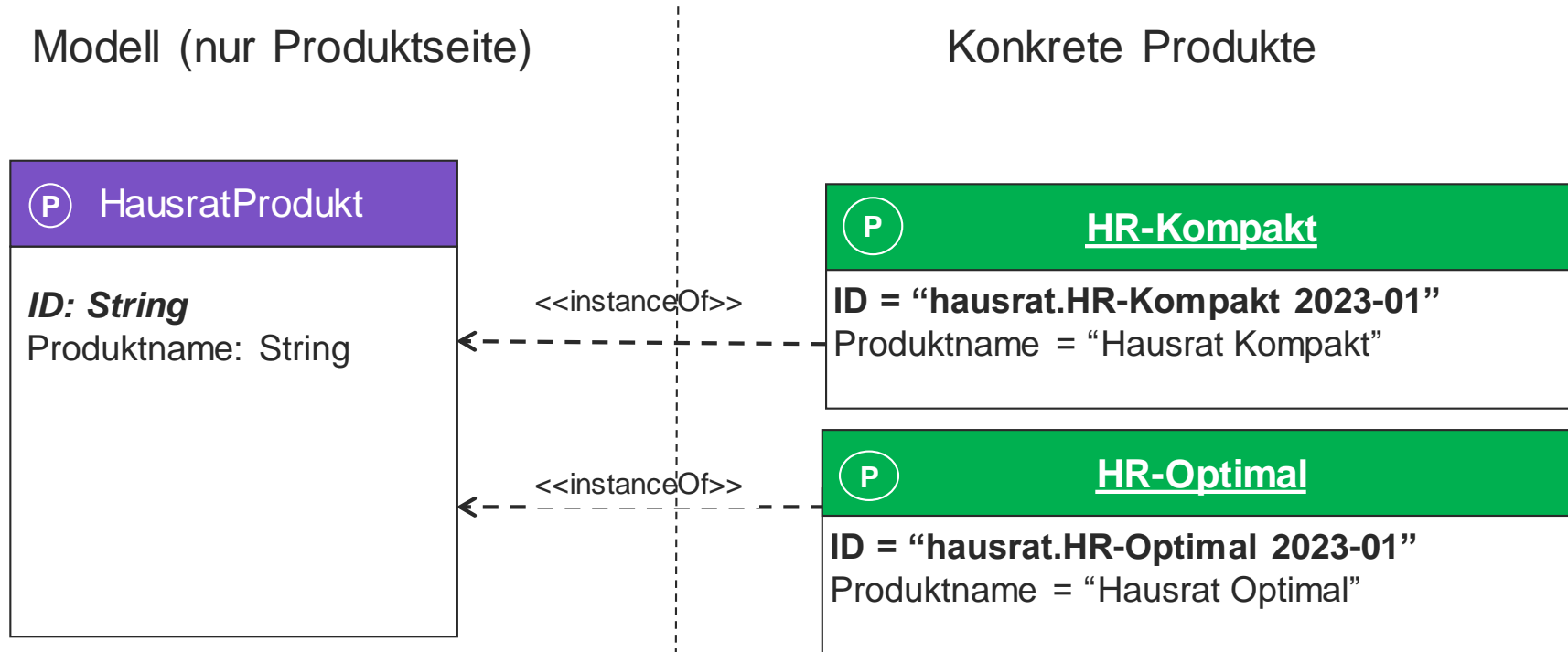
Konfigurierbare Vertragsattribute

Zugriff auf Produktdaten zur Laufzeit

Motivation

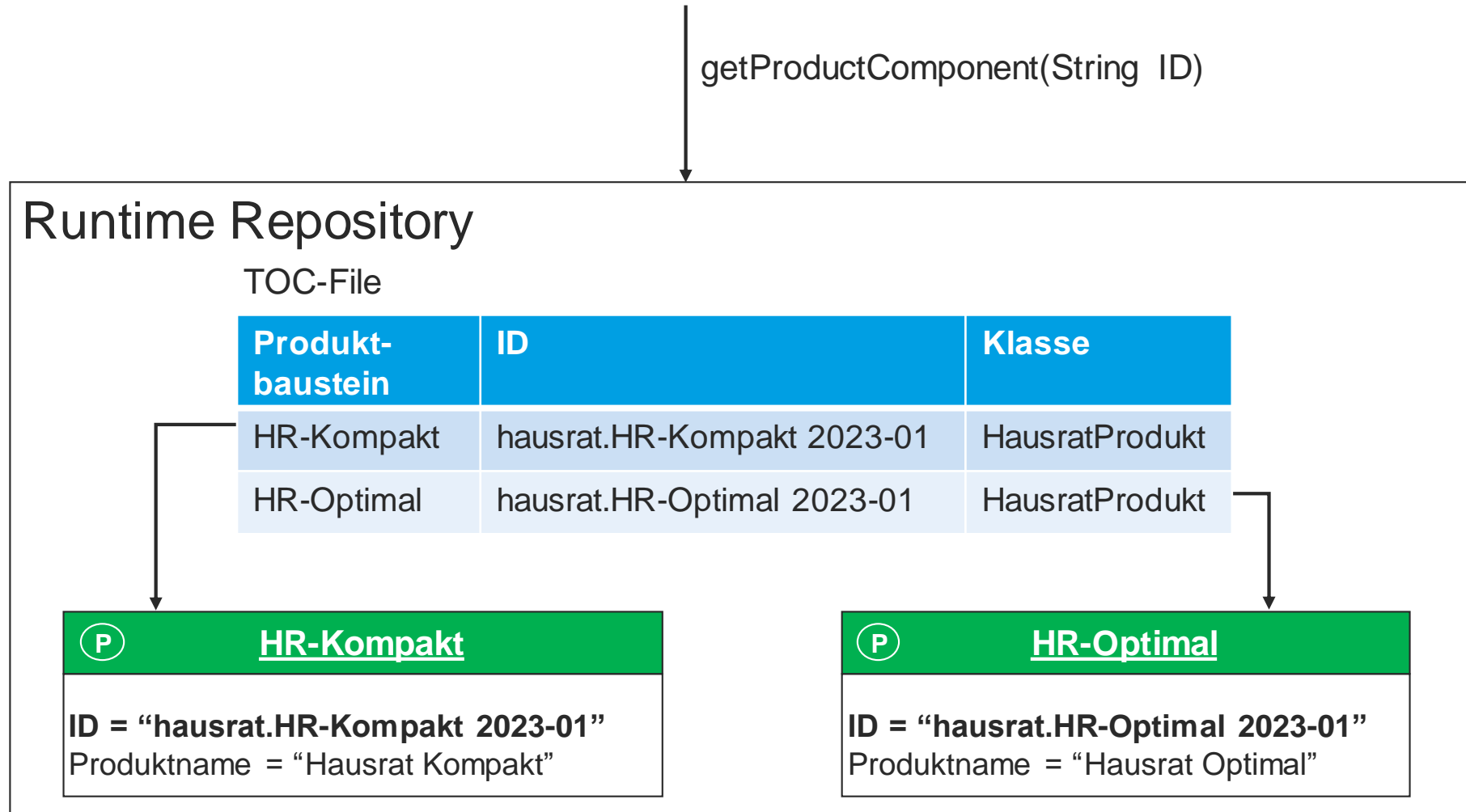
- Motivation:
- Auf Basis der Wohnfläche soll dem Kunden ein Vorschlag für die Versicherungssumme unterbreitet werden. Bisher wurde dazu die Wohnfläche mit dem fixen Betrag von 650,- € multipliziert.
- Die Fachabteilung möchte nun diesen Wert produktspezifisch definieren:
 - HR-Kompakt: 600 Euro
 - HR-Optimal: 900 Euro

Produktbausteine haben zur Laufzeit eine eindeutige ID.

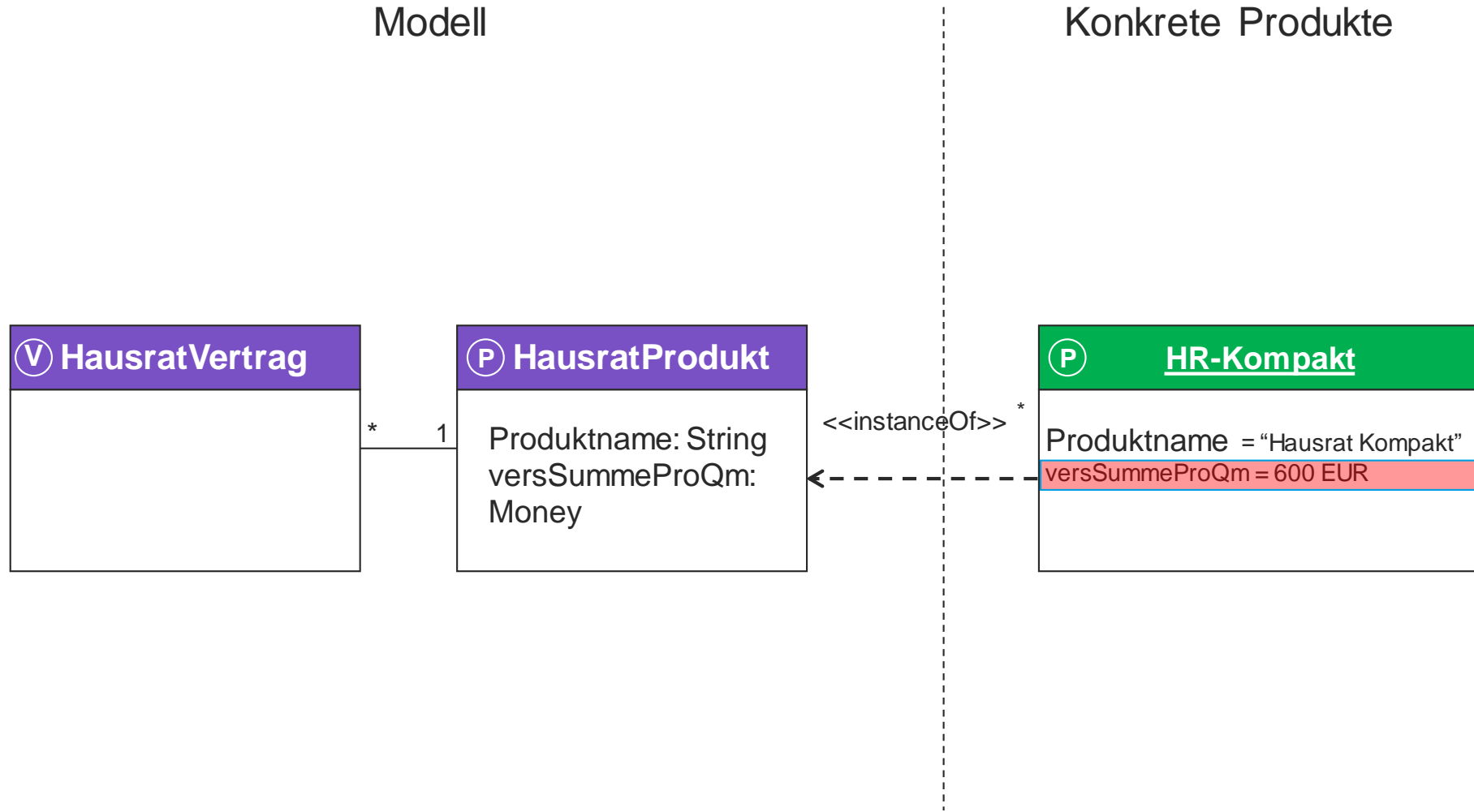


- Die ID ist eine grundlegende Eigenschaft eines Produktbausteins und muss nicht als Attribut angelegt werden.
- Sie wird automatisch beim Anlegen eines Produktbausteins erzeugt.
- Sie sollte bei einem „produktiven“ Produkt nie geändert werden

Der Zugriff auf Produktdaten zur Laufzeit erfolgt mit dem RuntimeRepository



Modellierung des Attributes

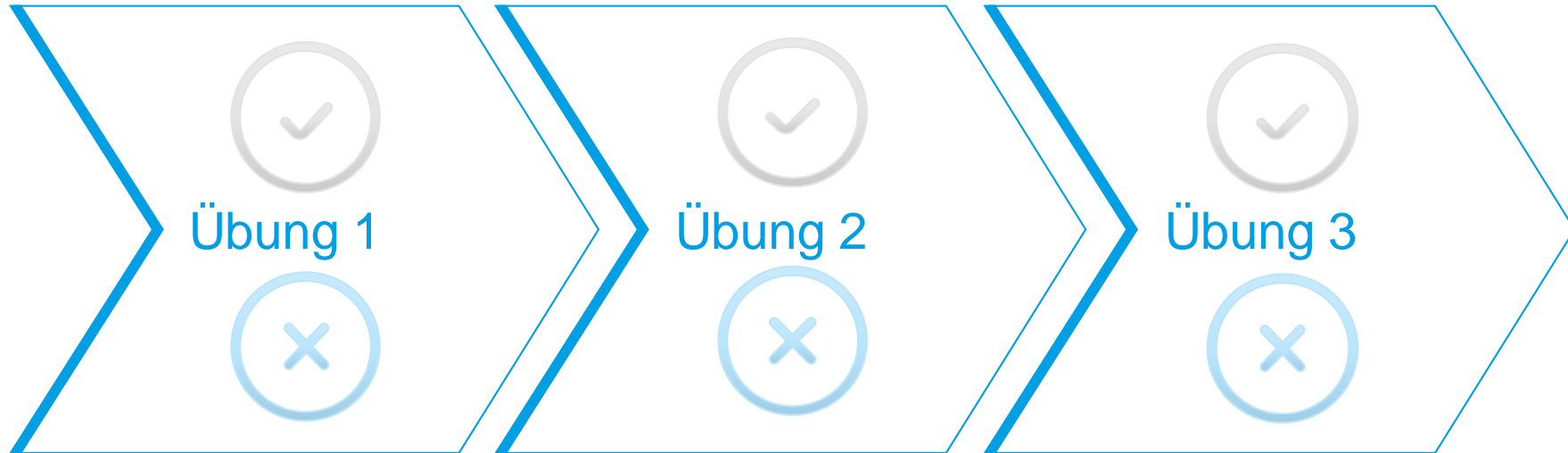


Demo: Zugriff auf Produktdaten

- Definition des Attributes „vorschlagVersSummeProQm“
- Eintragen der Produktwerte in den Produktbausteinen
- Ausgabe der Produktdaten
- Implementierung der Berechnung des Vorschlags für die Versicherungssumme
- Implementierung eines Testfalls

Übungen zu Kapitel III.C.4

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Agenda

01

Motivation

02

UML Refresh

03

Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Grundlagen

Beziehungen zwischen Produktbausteinen und Tabellen

Beispiel: Tarifzonentabelle

| PLZ-Von | PLZ-bis | Tarifzone |
|---------|---------|-----------|
| 17235 | 17237 | II |
| 45525 | 45549 | III |
| 59174 | 59199 | IV |
| 47051 | 47279 | V |
| 63065 | 63075 | VI |

Demo: Einführung Tarifzonentabelle

Anlegen der Struktur Tarifzonentabelle

Inhalt Tarifzonentabelle anlegen

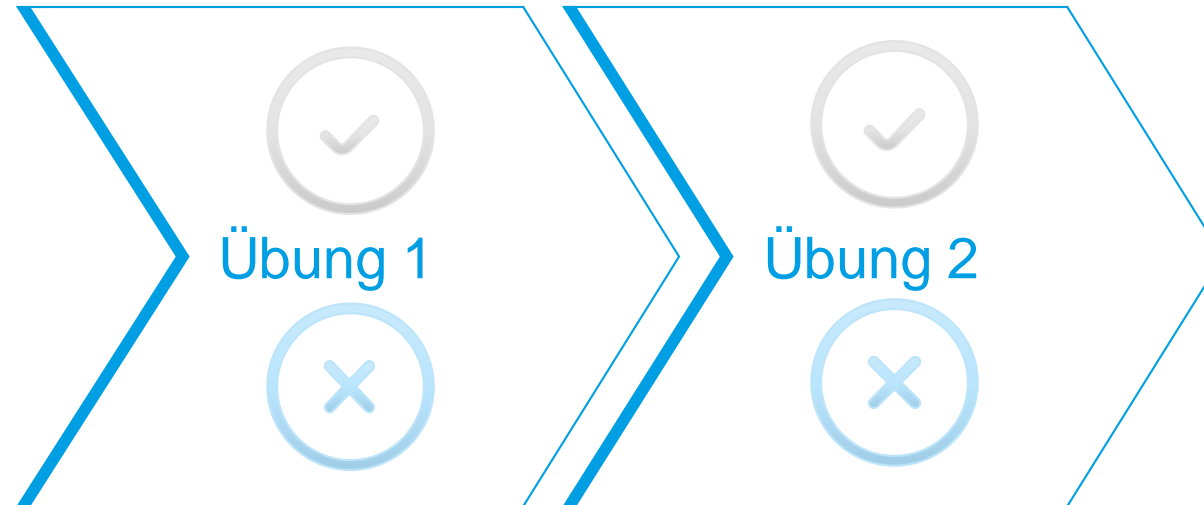
Generierten Code untersuchen

Verwendung der Tabelle zur Ermittlung der Tarifzone aus der PLZ im HausratVertrag

Schreiben eines Testfalls für die Ermittlung der Tarifzone.

Übungen zu Kapitel III.D.1

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01

Motivation

02

UML Refresh

03

Modellierung & Produktdefinition mit Faktor-IPS

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Grundlagen

Beziehungen zwischen Produktbausteinen und Tabellen

Beitragsberechnung für die Grunddeckungen

Berechnungsvorschrift

- Die Tabelle zum Produkt finden
- Ermittlung des Beitragssatzes aus der Tariftabelle
- Division der Versicherungssumme durch 1.000 und Multiplikation mit dem Beitragssatz
- Beitragssätze unterscheiden sich je Produkt!

Tariftabelle für die Grunddeckungen der Hausratprodukte

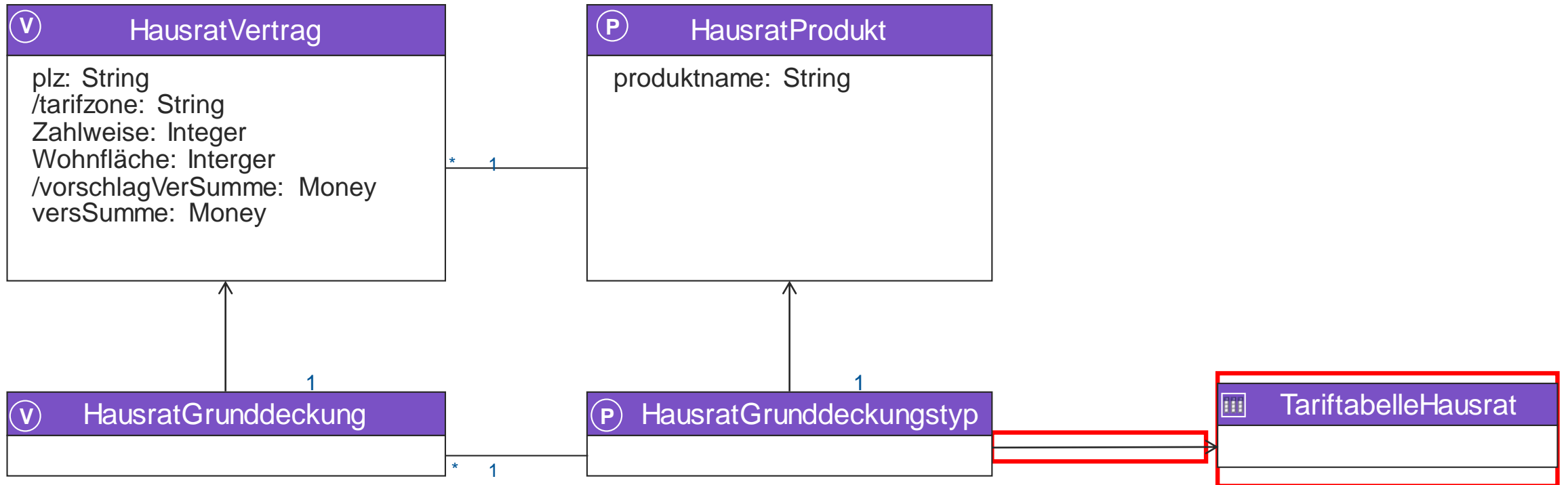
| Produkt | Tarifzone | Beitragssatz |
|------------|-----------|--------------|
| HR-Optimal | I | 0.8 |
| HR-Optimal | II | 1.0 |
| HR-Optimal | III | 1.2 |
| HR-Optimal | IV | 1.4 |
| HR-Optimal | V | 1.6 |
| HR-Optimal | VI | 1.8 |
| HR-Kompakt | I | 0.6 |
| HR-Kompakt | II | 0.8 |
| HR-Kompakt | III | 1.0 |
| HR-Kompakt | IV | 1.2 |
| HR-Kompakt | V | 1.4 |
| HR-Kompakt | VI | 1.6 |

Trennung der Tabelle nach Produkt

| Tabelle für Grunddeckung von HR-Kompakt | |
|---|--------------|
| Tarifzone | Beitragssatz |
| I | 0.6 |
| II | 0.8 |
| III | 1.0 |
| IV | 1.2 |
| V | 1.4 |
| VI | 1.6 |

| Tabelle für Grunddeckung von HR-Optimal | |
|---|--------------|
| Tarifzone | Beitragssatz |
| I | 0.8 |
| II | 1.0 |
| III | 1.2 |
| IV | 1.4 |
| V | 1.6 |
| VI | 1.8 |

Abbildung im Modell

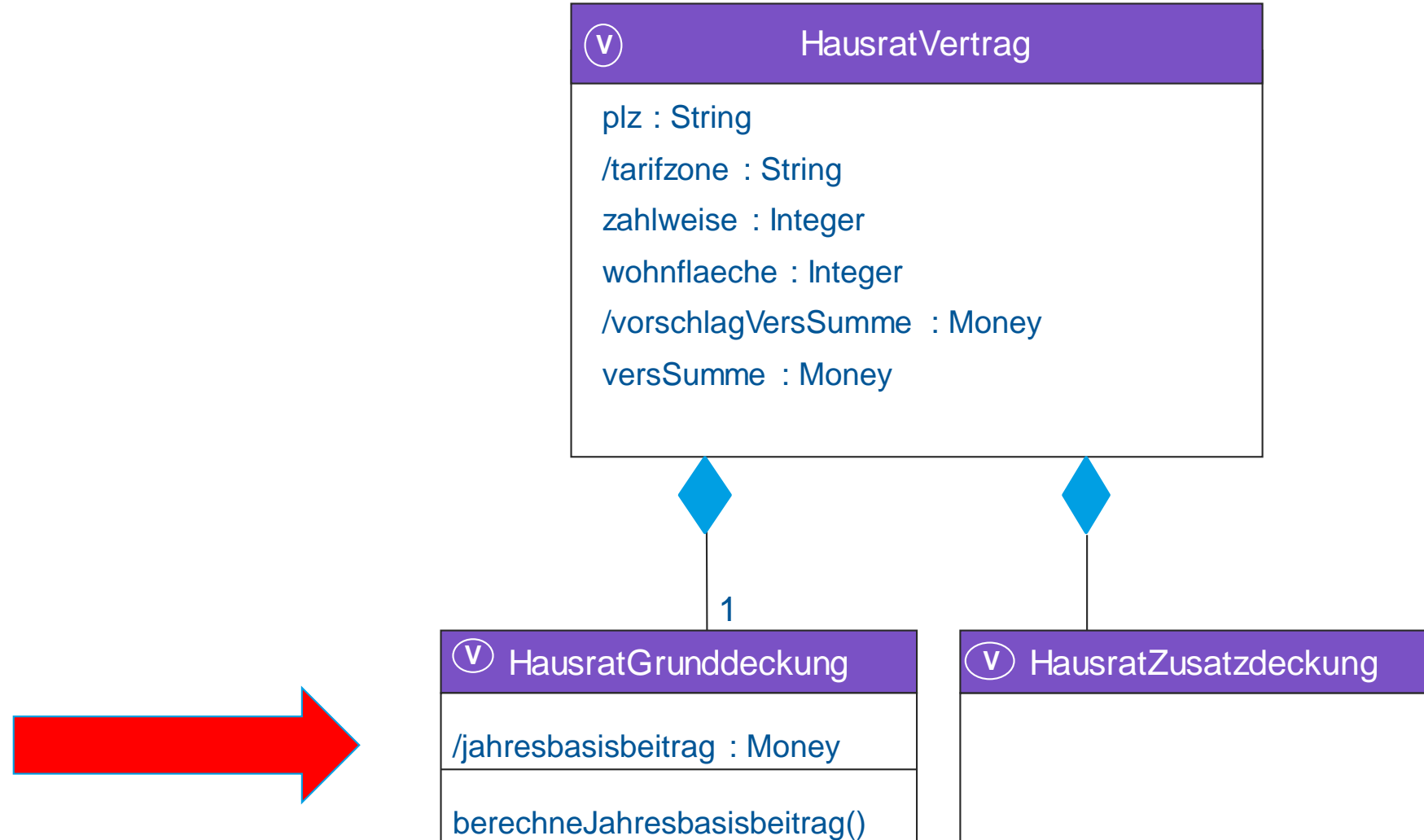


Ein Hausratgrunddeckungstyp verwendet eine Hausrattariftabelle (zur Beitragsberechnung)

Demo: Tariftabellen für die Hausratprodukte

- Anlegen der Tabellenstruktur für die Tariftabelle
- Anlegen der Beziehung zwischen Grunddeckungstyp und der Tariftabelle
- Wechseln in die Produktdefinitionsperspektive
- Anlegen der Tabelleninhalte

Umsetzung der Beitragsberechnung

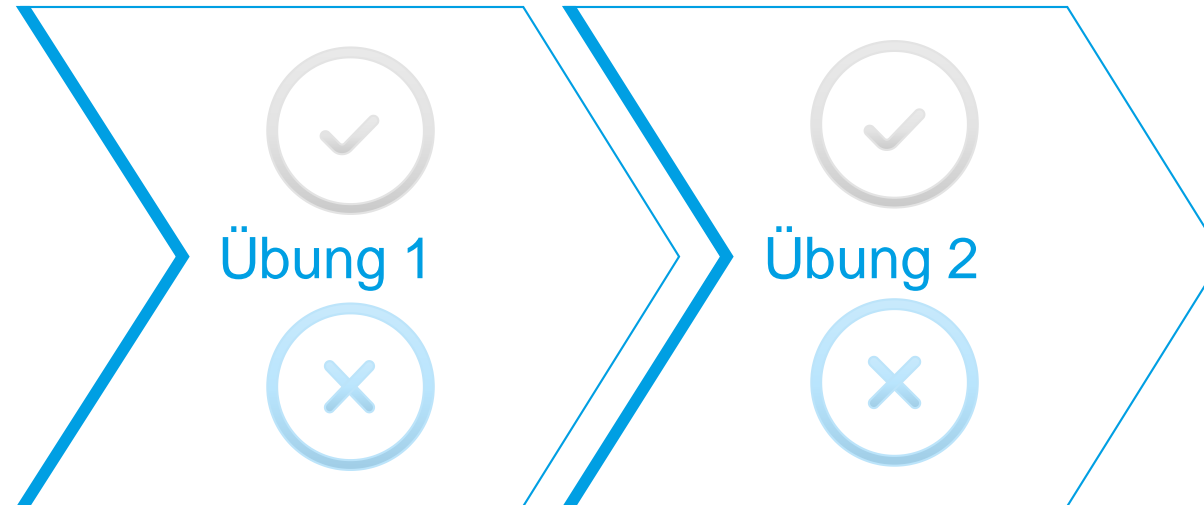


Demo: Umsetzung der Beitragsberechnung

- Anlegen des derived (cached) Attributes *jahresbasisbeitrag* in der Modellklasse Hausratgrunddeckung
- Definition der Methode `berechneJahresbasisbeitrag()` in der Modellklasse Hausratgrunddeckung
- Implementierung der Methode `berechneJahresbasisbeitrag()` in der Java Klasse Hausratgrunddeckung
- Testfall für die Methode implementieren

Übungen zu Kapitel III.D.2

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung


04 Customizing & Tools

Aufzählung Zahlweise: Model

Modell

| Zahlweise | |
|-----------|---------------|
| Id | Name |
| 1 | Jährlich |
| 2 | Halbjährlich |
| 4 | Quartalsweise |
| 12 | Monatlich |
| 0 | Einmalig |

Java



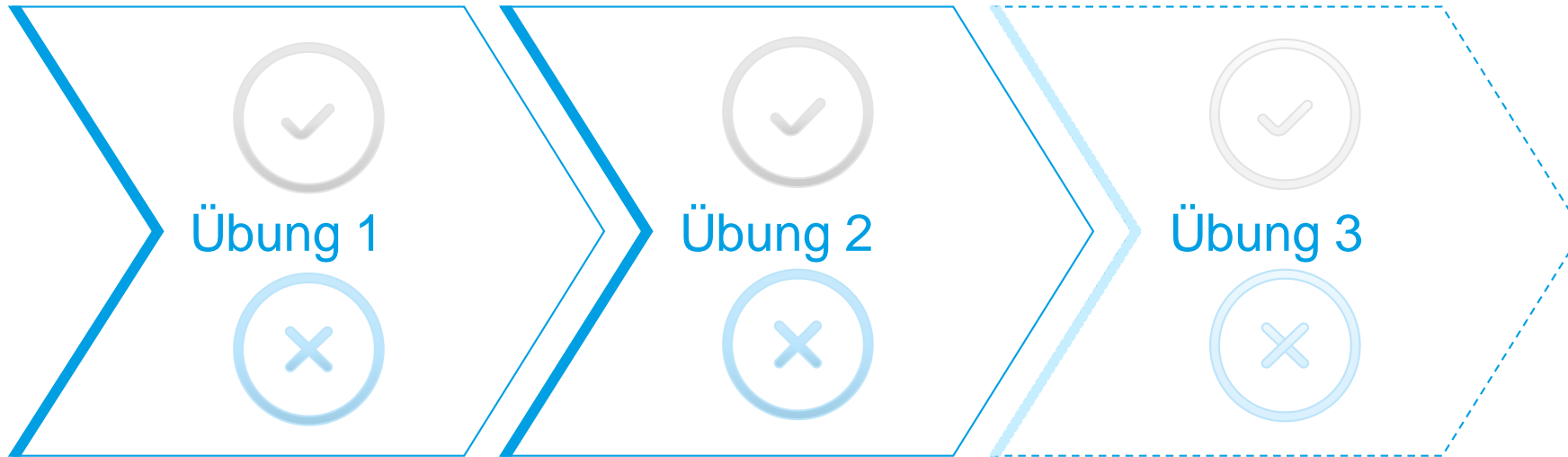
```
public enum Zahlweise {  
    /**  
     * @generated  
     */  
    JAEHRLICH(1, "jaehrlich"),  
    /**  
     * @generated  
     */  
    HALBJAEHRLICH(2, "halbjaehrlich"),  
    ...  
}
```

Demo: Aufzählung Zahlweise

- Aufzählungstyp Zahlweise anlegen
- Generierten Sourcecode anschauen
- Im HausratGrunddeckung das Attribut Zahlweise auf diesen Datentyp ändern
- Im HausratGrunddeckung berechnetes Attribut (bei jedem Aufruf) beitragsGemaessZahlweise anlegen
- In der Java-Klasse implementieren
- JUnit Test für die Beitragsberechnung erweitern.

Übungen 1-3 zu Kapitel III.E

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Aufzählung Risikoklasse: Modell und Produktseite

Modell

| ⓔ Riskiklasse | |
|---------------|------|
| ID | Name |
| | |
| | |
| | |

Java

```
public final class Riskiklasse {  
  
    /**  
     * @generated  
     */  
    private final String id;  
  
    /**  
     * @generated  
     */  
    private final String name;  
}
```

Produkt

| ⓔ Riskiklasse | |
|---------------|------------------------------------|
| ID | Name |
| 10 | Ständig bewohntes Einfamilienhaus |
| 20 | Ständig bewohntes Mehrfamilienhaus |
| 30 | Ständig bewohntes Ferienhaus |
| 40 | Zweitwohnung |

XML

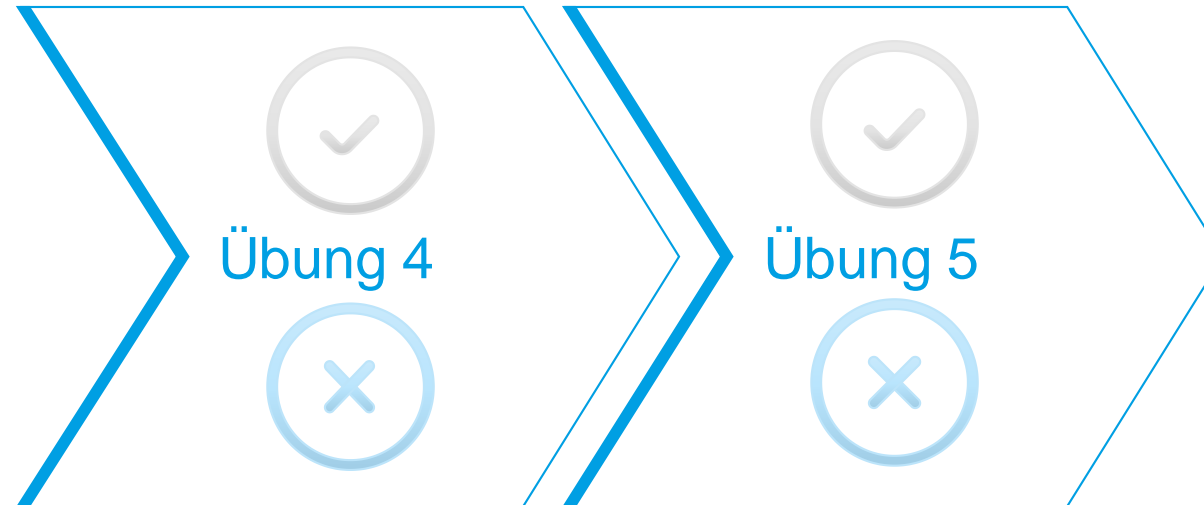
```
<?xml version="1.0" encoding="UTF-8"?>  
  
<EnumContent enumType="hausrat.Risikoklasse"  
  
...  
  
</EnumContent>
```


Demo: Aufzählung Risikoklasse

- Aufzählungstyp Risikoklasse anlegen
- Generierten Sourcecode und XML anschauen
- Junit-Test anlegen

Übungen 4 und 5 zu Kapitel III.E

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

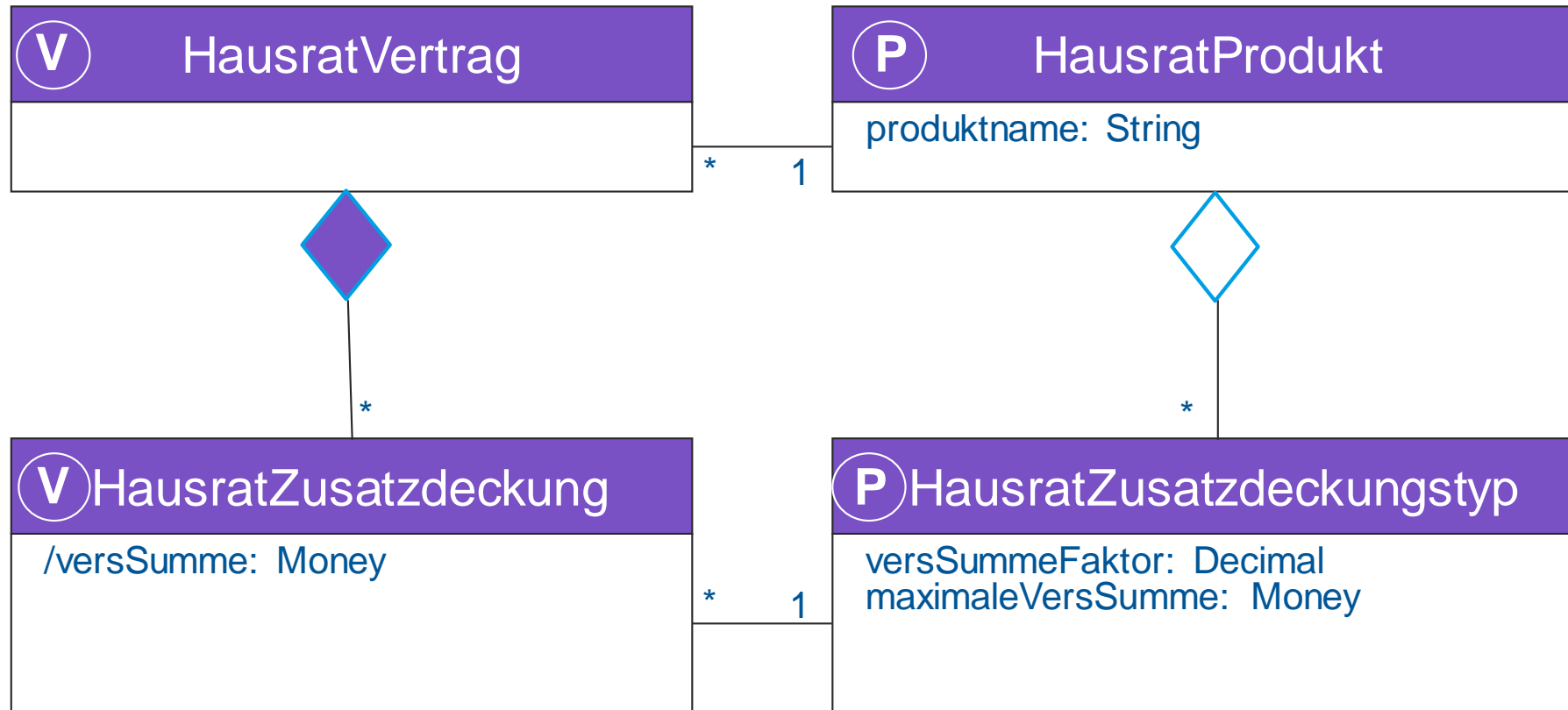
Fachliche Anforderungen

- Erweiterung des Modells, so dass Zusatzdeckungen durch die Fachabteilung hinzugefügt werden können, ohne dass das Modell geändert werden muss.
- Jede Zusatzdeckung verfügt über eine eigene Versicherungssumme und einen eigenen Jahresbasisbeitrag. Die Versicherungssumme ergibt sich aus der im Vertrag vereinbarten Summe.

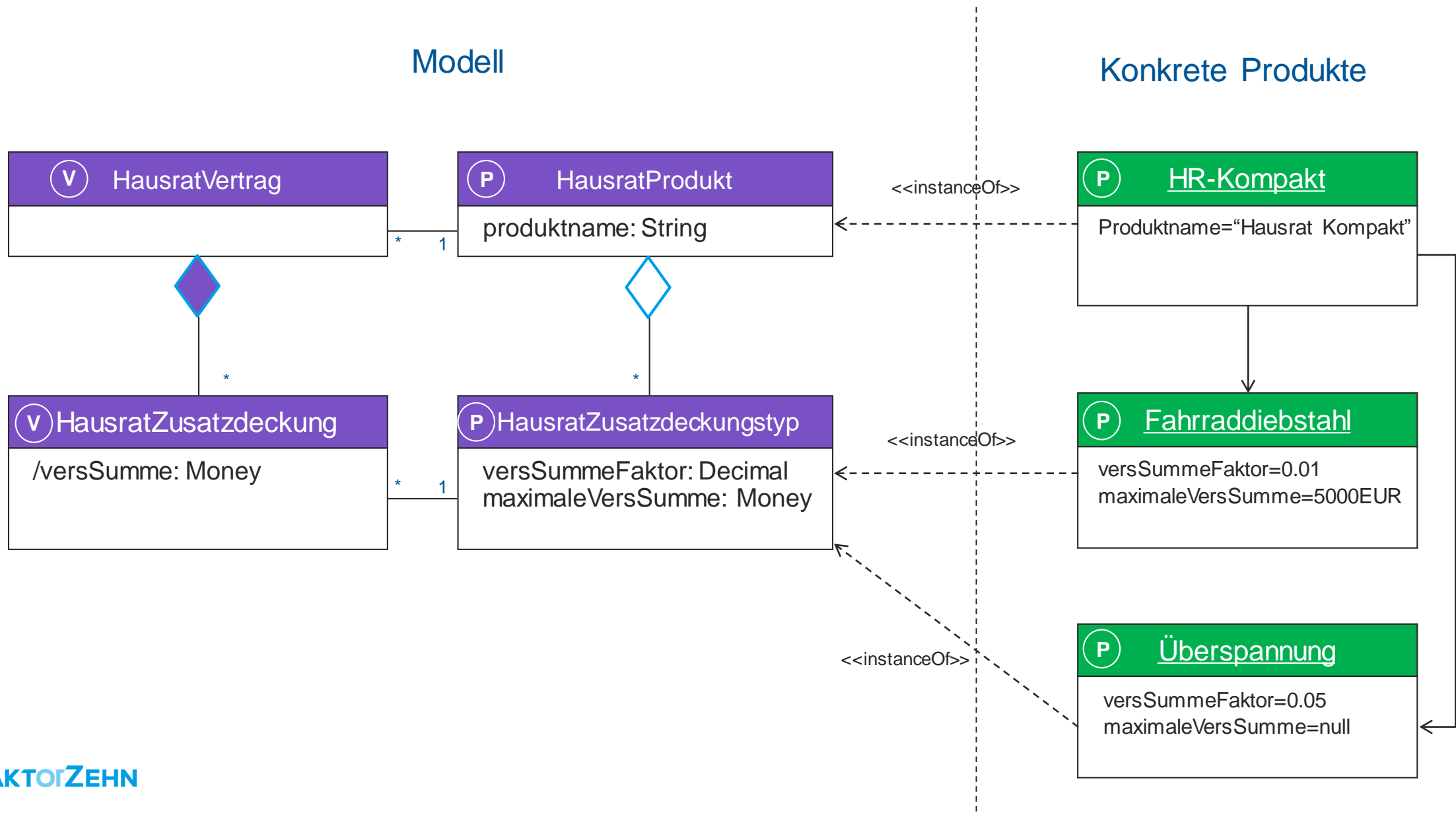
Beispiele:

| | Fahrraddiebstahl | Überspannung |
|--------------------------------------|---|--|
| Versicherungssumme der Zusatzdeckung | 1% der im Vertrag vereinbarten Summe, maximal 5000 Euro | 5% der im Vertrag vereinbarten Summe. Keine Deckelung |
| Jahresbasisbeitrag | 10% der Versicherungssumme der Fahrraddiebstahldeckung | 10 Euro + 3% der Versicherungssumme der Überspannungsdeckung |

Modell der Zusatzdeckungen



Modell der Zusatzdeckungen mit Produktbausteinen

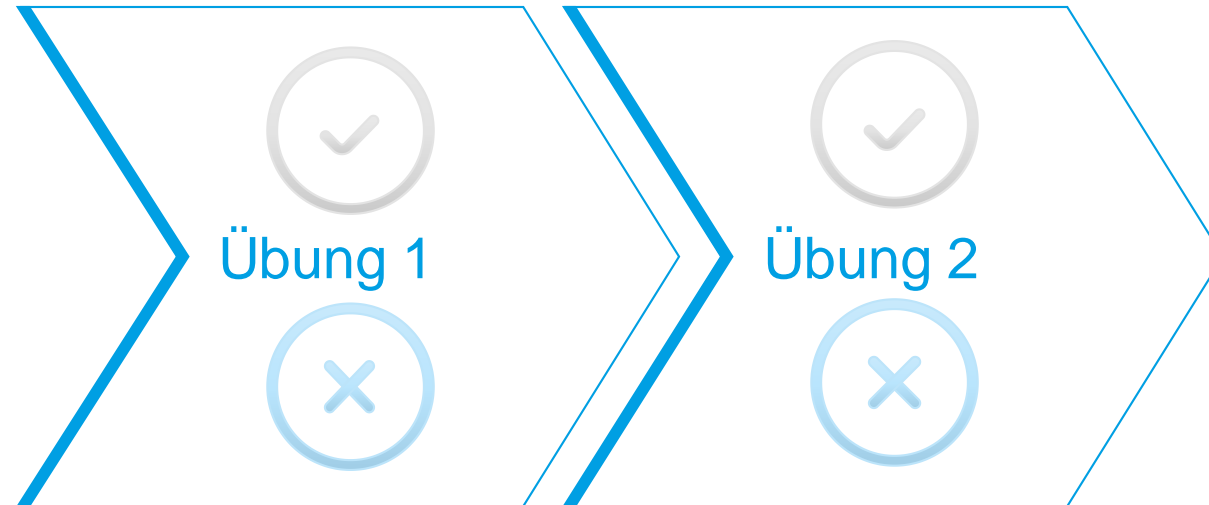


Demo: Anlegen der Zusatzdeckungen

| | HRD-Fahrraddiebstahl 2022-01 | HDR-Überspannung 2022-01 |
|--------------------|------------------------------|--------------------------|
| Bezeichnung | Fahrraddiebstahl | Überspannungsschutz |
| VersSummenFaktor | 0,01 | 0,05 |
| Maximale VersSumme | 5000 EUR | <null> |

Übungen 1 und 2 zu Kapitel III.F

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.

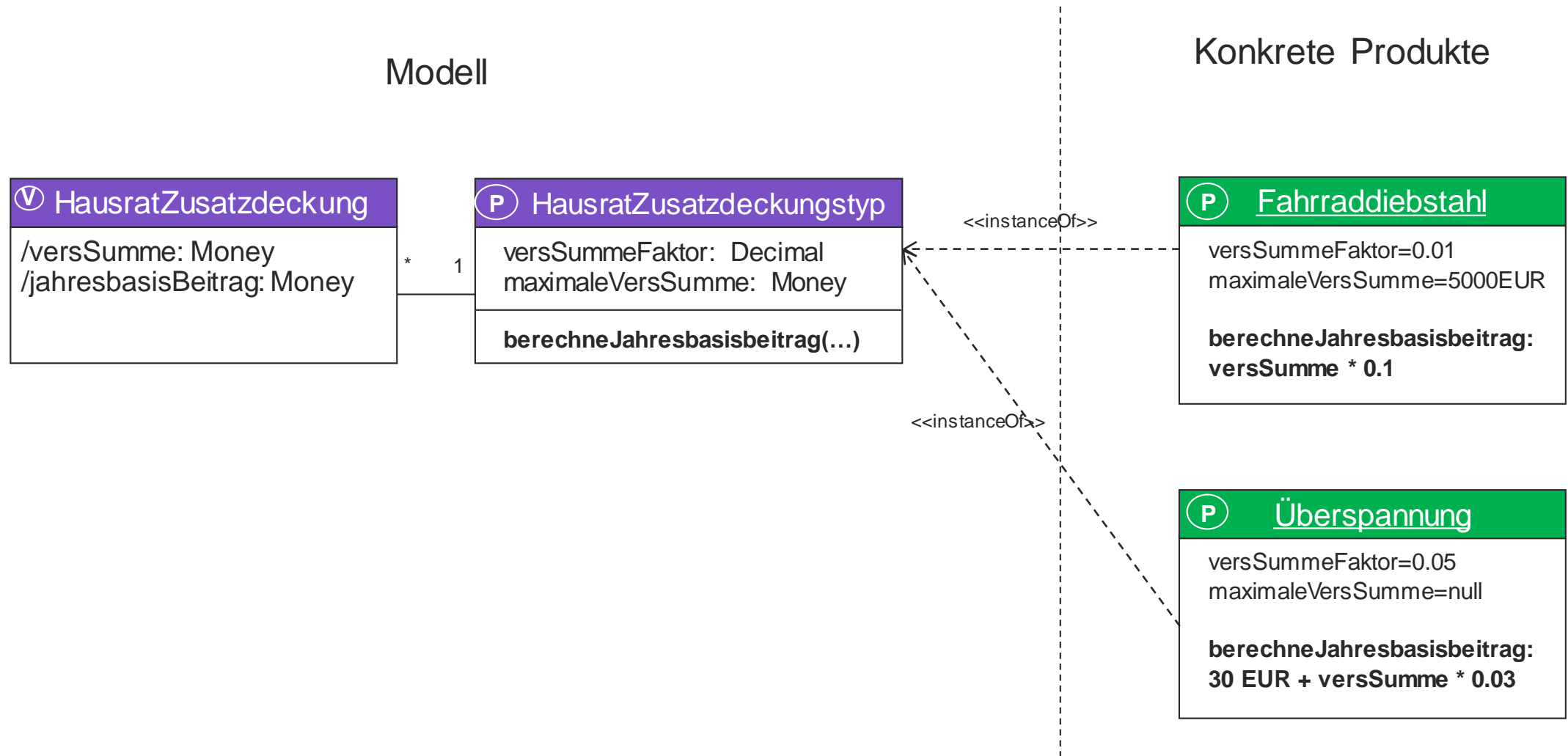


Berechnungen mit der Faktor-IPS Formelsprache

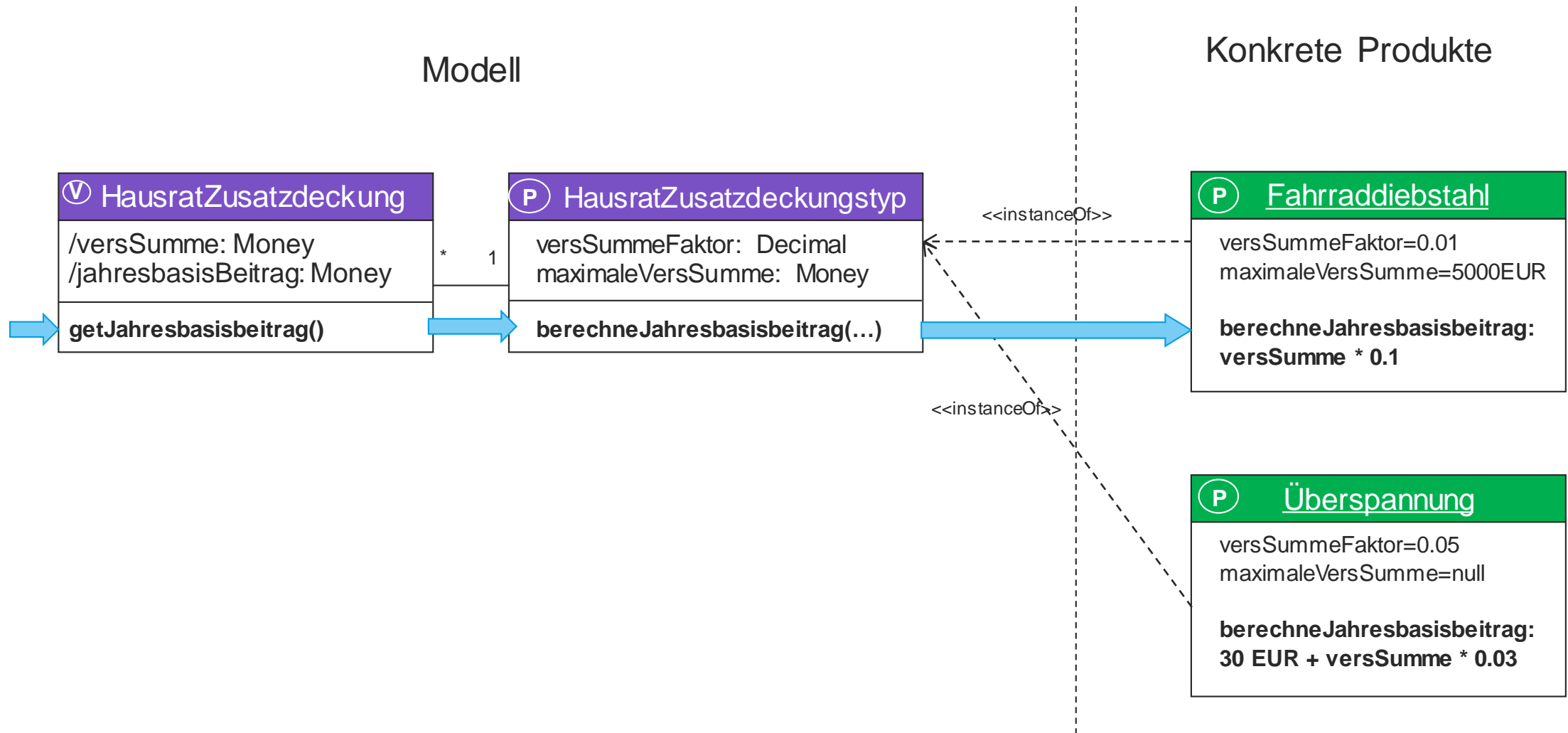
- Einfache Berechnungsvorschriften, die ein Fachbereich unabhängig von einer Anwendungsentwicklung implementieren möchte, können in Faktor-IPS mit Formelausdrücken angegeben werden.
- Die Formelsprache ist an die Formelsprache von Excel angelehnt.
- Um Formeln in einem Produktbaustein angeben zu können, muss in der Produktklasse des Bausteines eine Formelsignatur angegeben werden.

| | Fahrraddiebstahl | Überspannung |
|--------------------|--------------------------|--|
| Jahresbasisbeitrag | $\text{versSumme} * 0,1$ | $30 \text{ EUR} + \text{versSumme} * 0,03$ |

Berücksichtigung der Beitragsberechnung im Modell



Ablauf der Berechnung des Jahresbasisbeitrags

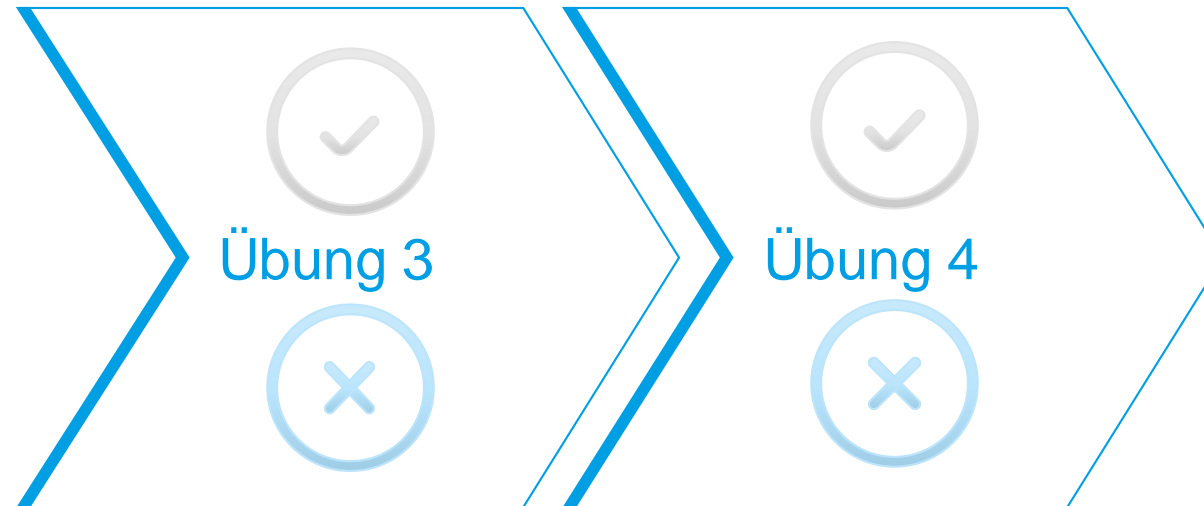


Demo: Beitragsberechnung für die Zusatzdeckungen

- Neues Attribut Jahresbasisbeitrag an der Zusatzdeckung
- Formelsignatur am Zusatzdeckungstyp
- Delegation von der Zusatzdeckung zum Zusatzdeckungstyp
- Berechnungsvorschriften in den Zusatzdeckungen
 - Fahrraddiebstahl 10% der Versicherungssumme
 - Überspannung 30EUR + 3% der Versicherungssumme

Übungen 3 und 4 zu Kapitel III.F

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Formelausführung zur Laufzeit

- Java
 - Subklassen für Produktbausteine
 - Müssen mit XML-Produktdaten deployt werden und auf dem Java-Classpath liegen
- Groovy
 - Formelcode im XML
 - Getrenntes Deployment möglich
 - RuntimeRepository benötigt GroovyFormulaEvaluator
 - Interpretation bei erstem Aufruf etwas langsamer

**Faktor-IPS Code
Generator Setting
“Formula Compiling”
Subclass/XML/Both**

Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Die Vertragsdaten sollen plausibilisiert werden

Aus der Fachabteilung kommen folgende Anforderungen für Plausibilisierungen:

- Die Wohnfläche am HausratVertrag darf nur die im jeweiligen Produkt konfigurierten Werte annehmen.
- Die Postleitzahl am HausratVertrag muss exakt 5 Ziffern enthalten.
- Eine HausratGrunddeckung darf nur am HausratVertrag eingebunden sein, wenn beide Kompakt oder beide Optimal sind.
- Bei monatlicher Zahlweise darf der Beitrag gemäß Zahlweise nicht unter 10€ betragen.

Die Plausibilisierungen sollen eine weitere Bearbeitung wie z.B. eine Reaktion von Endkunden auf Fehlermeldungen unterstützen.

Aspekte der Plausibilisierung

User-Interface
Zahlweise:
Monatlich 



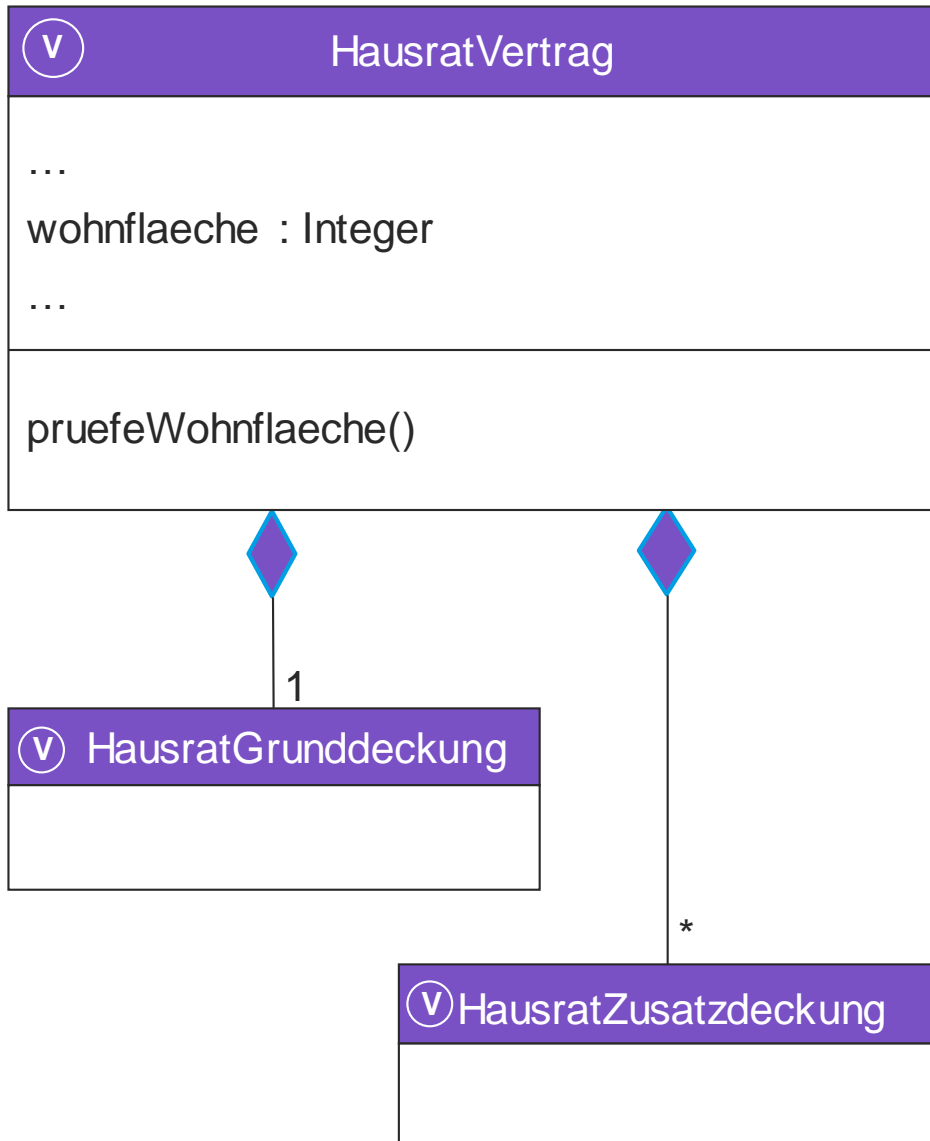
V HausratVertrag

plz : String
/tarifzone : String
zahlweise : Integer
wohnflaeche : Integer
/vorschlagVersSumme
versSumme : Money

Im Zusammenspiel zwischen Vertragsklasse sieht man drei Aspekte der Plausibilisierung:

- Prüfung
 - Regeln sollten innerhalb des Fachmodells geprüft werden.
 - Die Prüfung muss von außerhalb des Fachmodells explizit angestoßen werden (z. B. nach Abschluss einer Eingabe)
- Fehlermeldung
 - Das Ergebnis einer Plausibilisierung muss verständlich und nachvollziehbar sein.
- Auskunft
 - Informationen über Plausibilisierungen stehen Nutzern des Fachmodells zur Verfügung:
 - Aufbau von Oberflächen (z.B. Aufbau von Drop-Downs)
 - Clientseitige Vorab-Plausibilisierungen

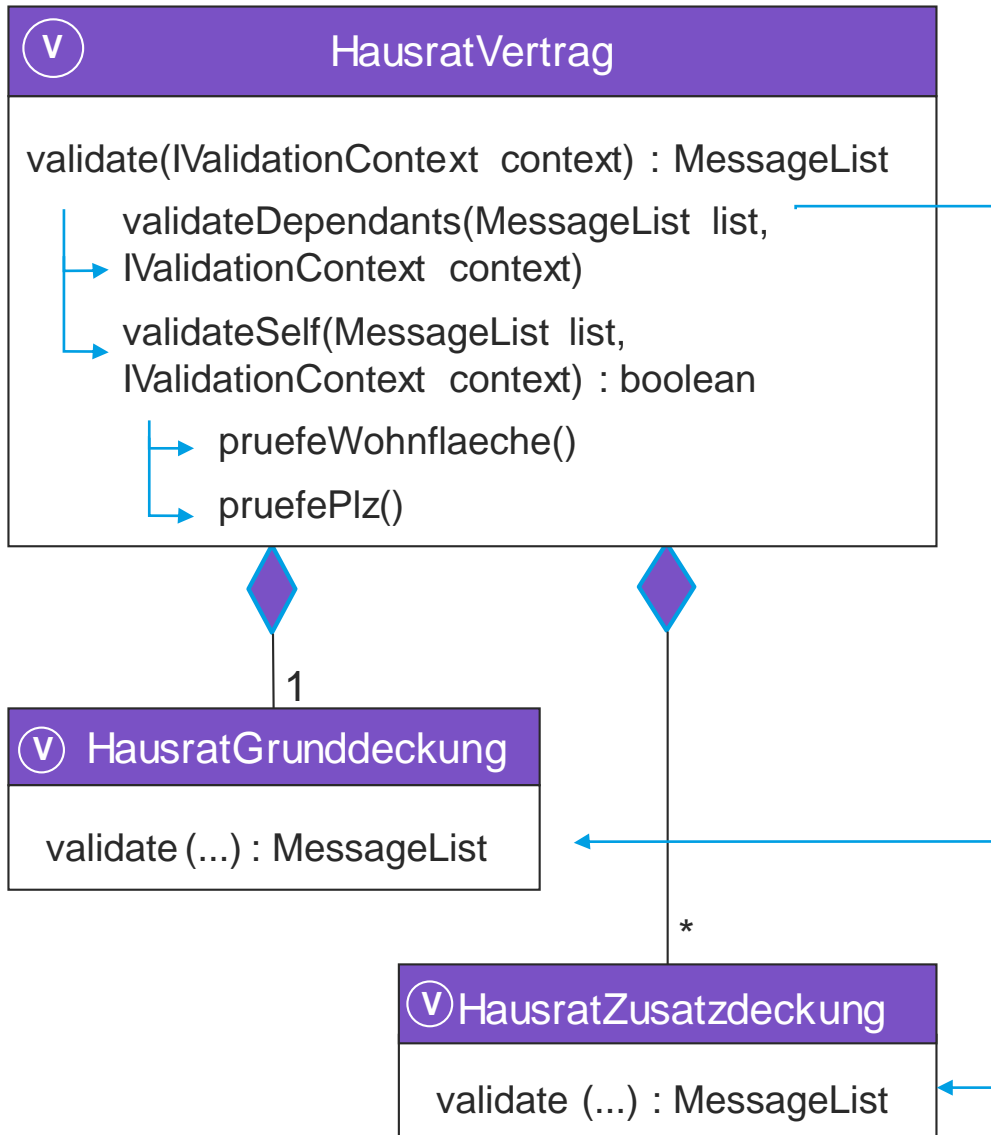
Prüfungen in Faktor-IPS



Prüfung

- Prüfungen erfolgen auf der Vertragsseite.
- Faktor-IPS bildet Prüfungen als Regeln z.B. `pruefeZahlweise` ab.
- Wertebereichsregeln können vollständig generiert werden.
- Die Logik einer Regel ist standardmäßig im Java Code zu implementieren.
- Faktor-IPS generiert zwei Methoden pro Regel:
 - eine zum Prüfen (kann bearbeitet werden)
 - eine für das Erzeugen einer Message

Prüfung eines Vertragsaggregats



- Eine Prüfung wird an einem beliebigen Vertragsobjekt innerhalb des Aggregats mit dem Aufruf von `validate()` explizit gestartet.
- Bei einer Prüfung werden durch die Codegenerierung automatisch auch alle Kinder des Vertragsobjekts berücksichtigt, in dem automatisch deren `validate()`-Methoden gerufen werden.
- Alle Messages werden dabei in der `MessageList` gesammelt.
- Der `IValidationContext` enthält Informationen wie die Sprache von Fehlermeldungen.

Prüfungen in Faktor-IPS und Produktinformationen

| V | HausratVertrag |
|-----|-----------------------|
| ... | |
| | wohnflaeche : Integer |
| | zahlweise |
| ... | |
| | pruefeWohnflaeche() |
| | pruefeZahlweise() |

| P | HausratProdukt |
|-----|------------------------------------|
| ... | |
| | rangeForWohnflaeche : IntegerRange |
| | allowedValuesForZahlweise : List<> |
| ... | |

Konfiguration von Regeln über Produktinformationen

- In Regeln können Produktattribute ausgewertet werden
- Wertebereichsregeln können über die Angabe einer Wertemenge im Produktbaustein konfiguriert werden (rangeForWohnflaeche und allowedValuesForZahlweise)
- Regeln können produktseitig aus- und eingeschaltet werden (in Anpassungsstufen)
- Die Bedingung kann auch über eine Formel abgebildet werden, die dann in der Regel aufgerufen wird.

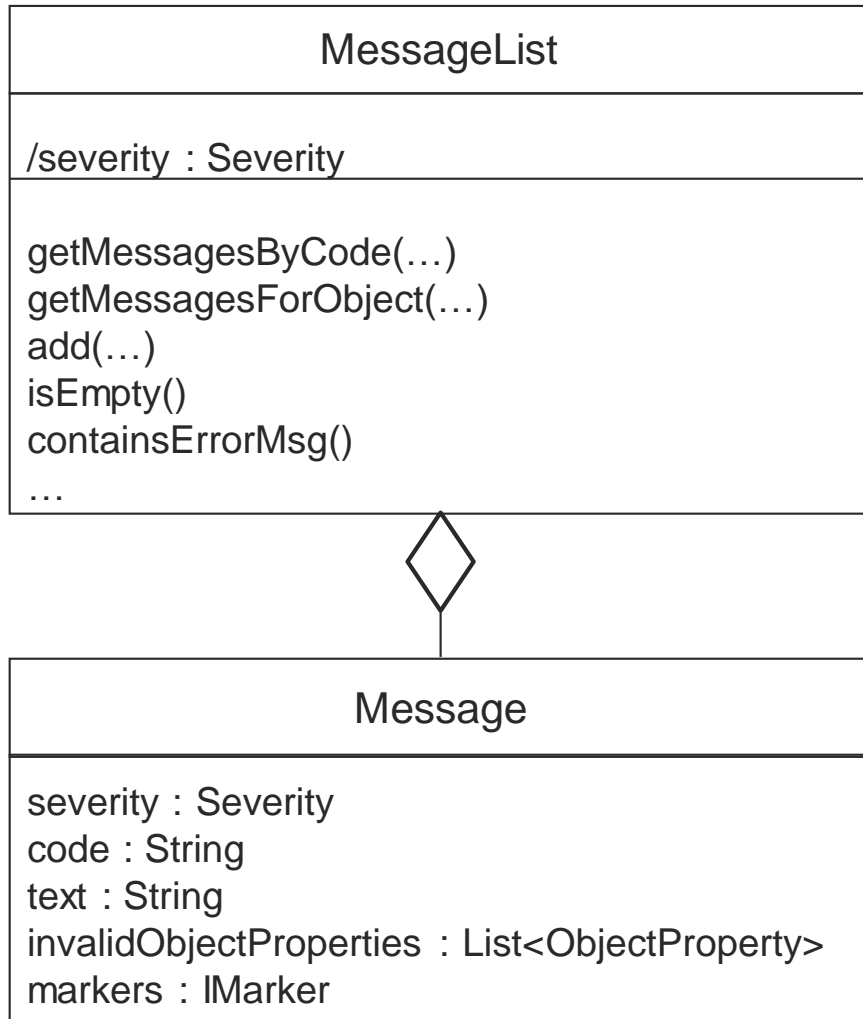
Fehlermeldungen mit den Klassen Message und MessageList

| Message |
|--|
| severity : Severity code : String text : String invalidObjectProperties : List<ObjectProperty> markers : IMarker |

Message

- Die Message enthält einen Schweregrad, einen Fehlercode sowie einen Fehlertext. Der Fehlertext kann Parameter enthalten, die bei der Erstellung einer Message verwendet werden (z.B. Angabe der erlaubten Werte)
- Die invalidObjectProperties enthalten eine Referenz auf fehlerhafte Vertragsobjekte und die jeweilige Eigenschaft (z.B. das Attribut Zahlweise)
- Die markers kategorisieren Messages nach bestimmten, projektspezifischen Merkmalen

Fehlermeldungen mit den Klassen Message und MessageList



MessageList

- Der Schweregrad einer MessageList ergibt sich aus dem höchsten Schweregrad der enthaltenen Messages
- Mit den Methoden `getMessagesByCode` bzw. `getMessagesForObject` kann gezielt nach Messages gesucht werden (z.B. für die Anzeige von Fehlern zu einem Vertragsobjekt).
- Die Methoden `isEmpty()` und `containsErrorMsg()` unterstützen die Auswertung einer MessageList

Auskunftsfunktionen in Faktor-IPS für Vertragsattribute

| HausratVertrag |
|--|
| zahlweise : Zahlweise wohnflaeche : Integer |
| getAllowedValuesForZahlweise() getRangeForWohnflaeche() |

**Ab Faktor-IPS 22.6 optional einheitlich
getAllowedValuesFor[Attributname]()**

| HausratProdukt |
|--|
| defaultValueZahlweise defaultValueWohnflaeche |
| getAllowedValuesForZahlweise() getRangeForWohnflaeche() |

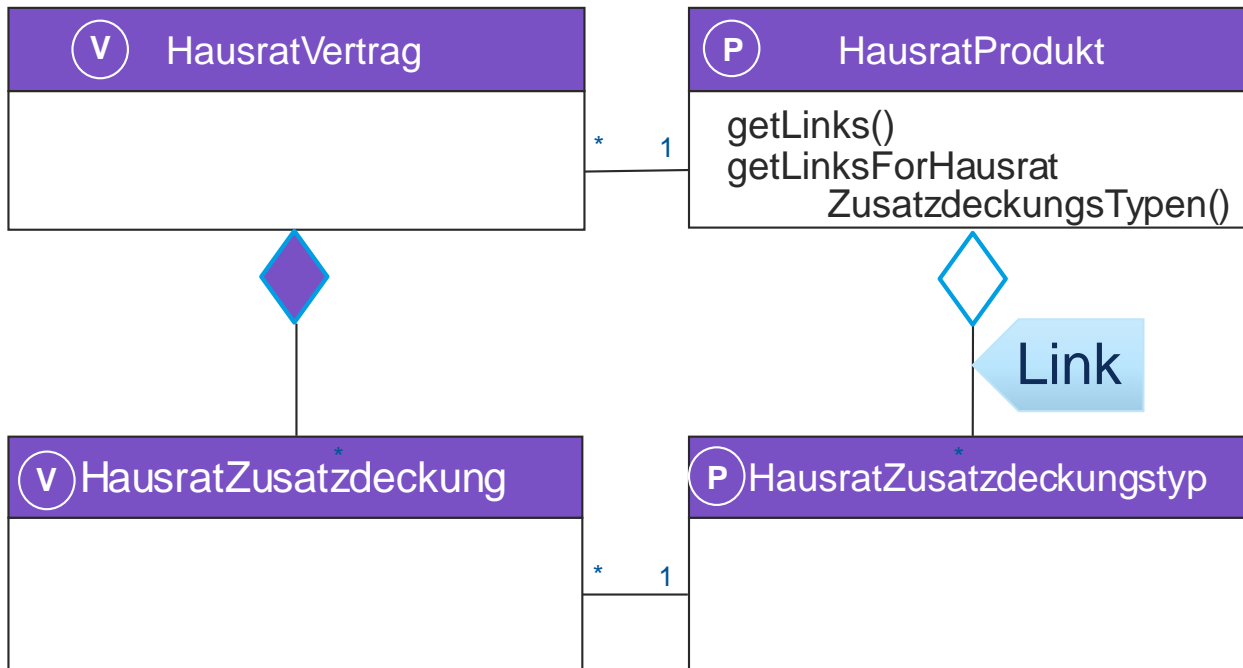
Allgemein

- Faktor-IPS generiert Auskunftsmethoden für Modellobjekte.

Vertragsattribute

- Erlaubte Werte werden je nach Art des Wertebereichs an der Vertragsklasse (bzw. bei konfigurierbaren Vertragsattributen an der Produktbausteinklasse) beauskunftet:
 - Bereich (z.B. Wohnfläche)
`getRangeFor[Attributname](IValidationContext)`
 - Aufzählung (z.B. Zahlweise)
`getAllowedValuesFor[Attributname](IValidationContext)`
- Faktor-IPS generiert den Code für Wertebereichsregeln und ruft im Code diese Auskunftsmethoden auf.
- Der Defaultwert für ein konfiguriertes Vertragsattribut kann aus der Produktbausteinklasse gelesen werden.

Auskunftsfunktionen in Faktor-IPS für Vertragsbeziehungen



Vertragsbeziehungen

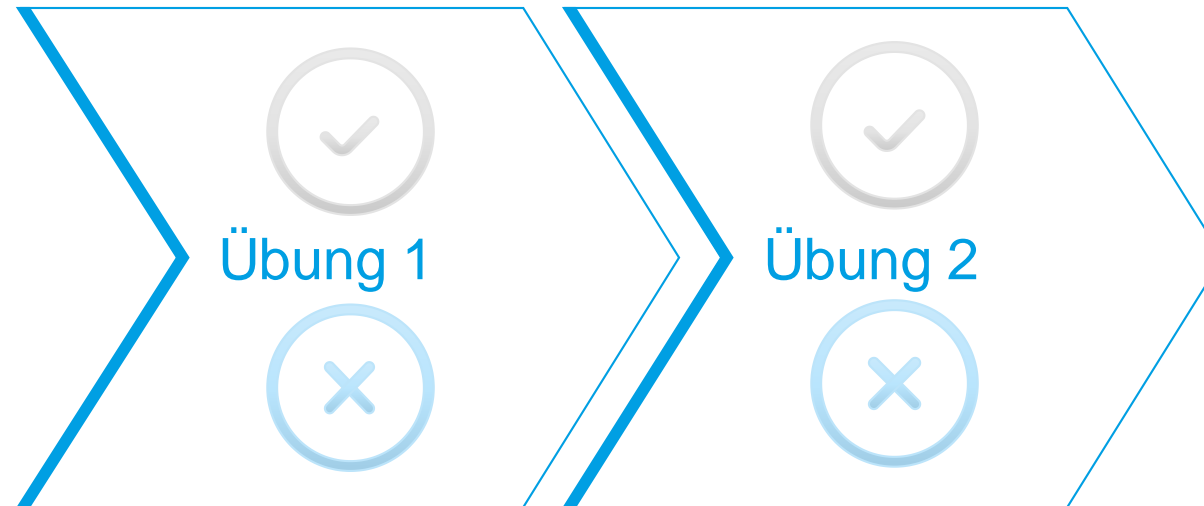
- Produktbausteinclassen geben in den Methoden `getLinks()` bzw. für eine einzelne Beziehung `getLinksFor[Beziehungsname]()` eine **Collection** von Links aus.
- Ein Link entspricht einer Beziehung von einem Produktbaustein zu einem anderen, z.B. von HausratKompakt zu Fahrraddiebstahl.
 - Der Link enthält Quell und Zielbaustein sowie den Namen der Beziehung.
 - Der Link enthält die minimale, maximale und Standardkardinalität.
- Diese Informationen können bei Regeln über erlaubte Einschlüsse und zur Steuerung von Oberflächen ausgewertet werden.

Demo: Validierungsregeln

- Wertebereichsregel
- Eigenständige Regel
- Meldungstext
- Invalid objects
- Generierter Code

Übungen zu Kapitel III.G

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Generische Validierung

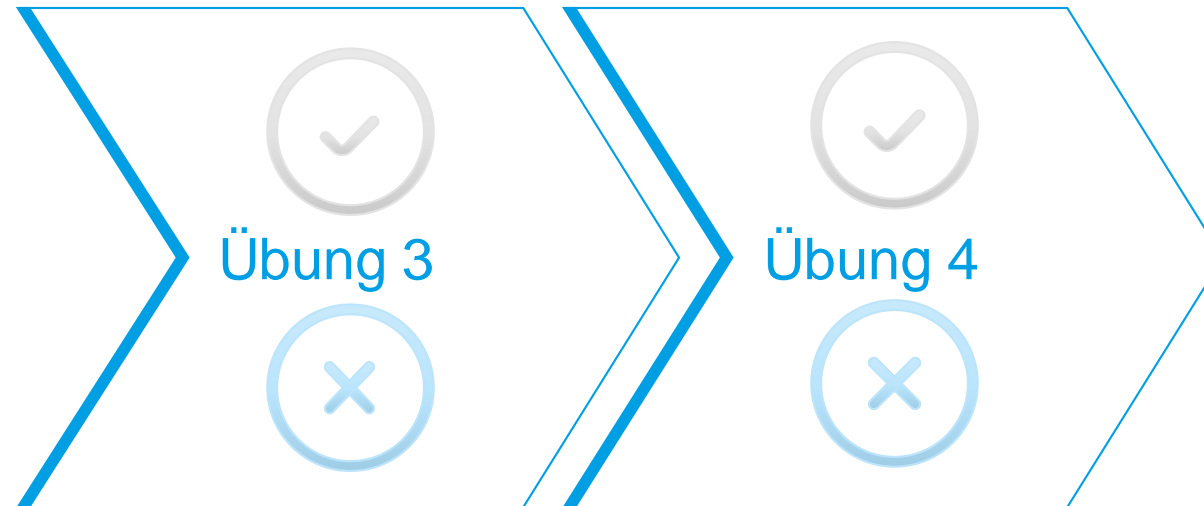
- Validierung von Wertebereichen mit einheitlichen Fehlermeldungen ohne individuellen Code
- Über ValidationContext konfigurierbar
- Nur noch Prüfungen die über einfache Wertebereichsprüfung hinausgehen müssen implementiert werden.

Demo: Validierungsregeln

- Generische Validierung
- Label

Übungen zu Kapitel III.G

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

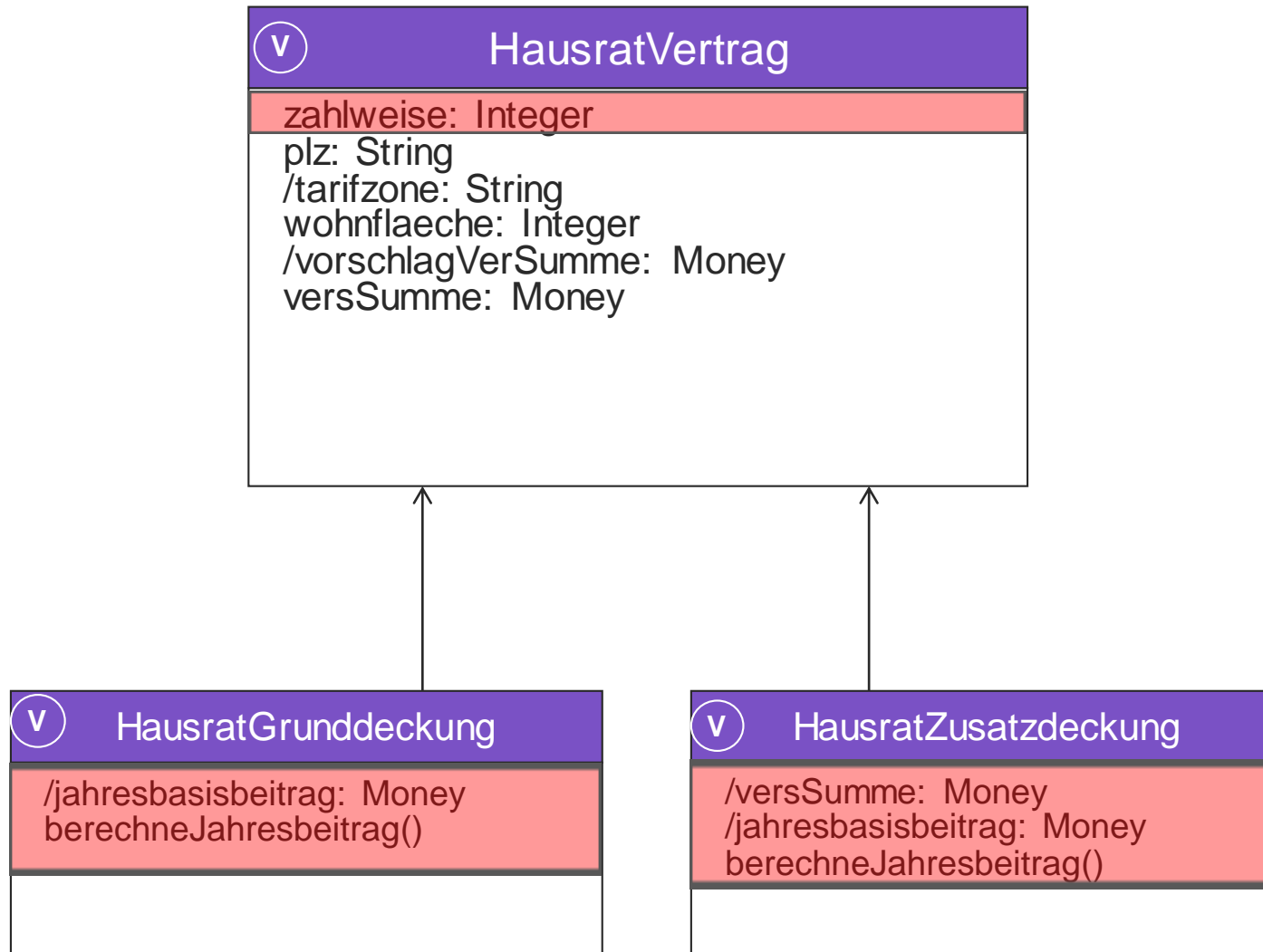
Plausibilisierungen

Vererbung

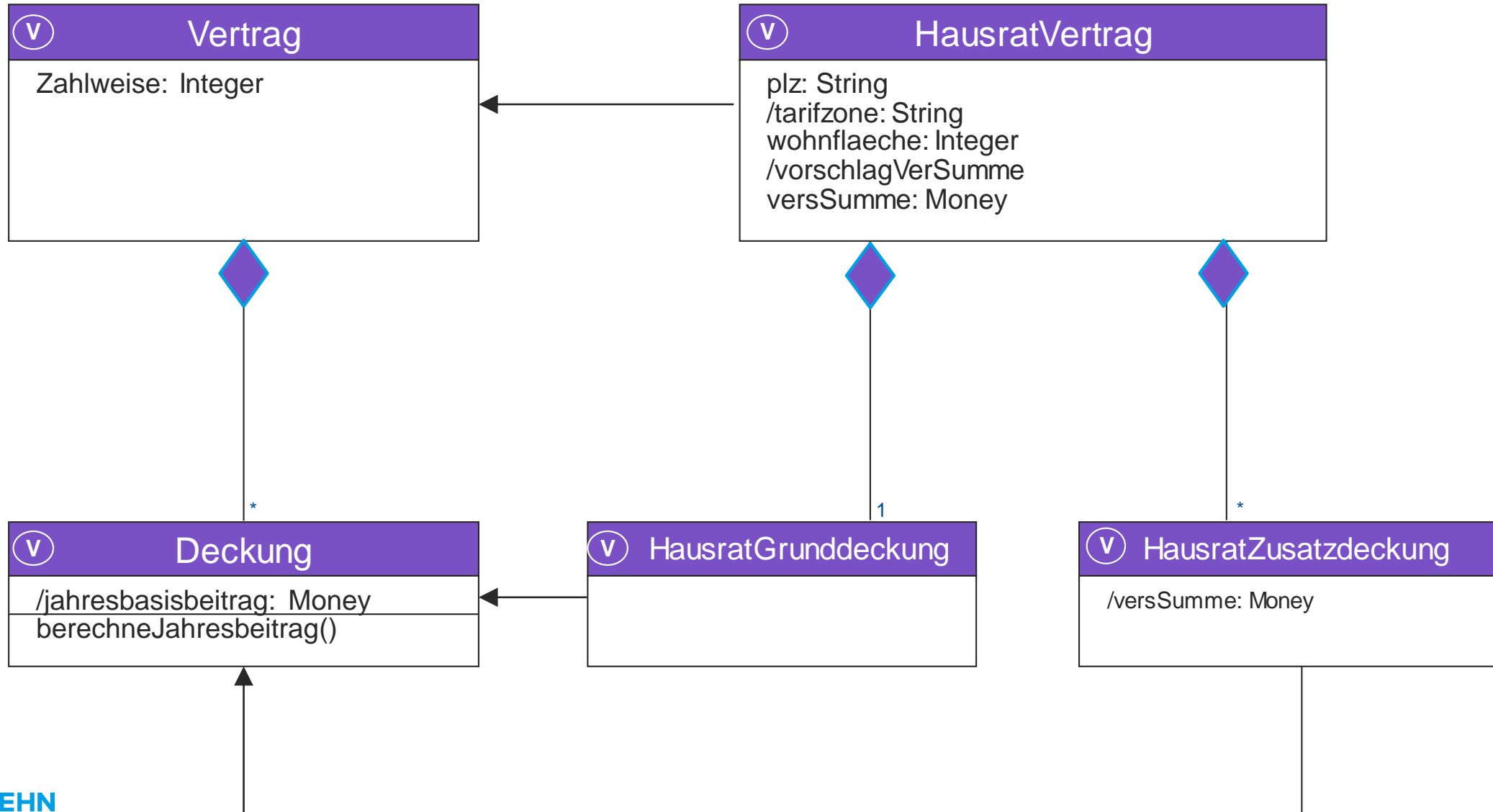
Testunterstützung

04 Customizing & Tools

Aktuelles Modell



Modell mit spartenübergreifenden Basisklassen



Demo: Einführung der Basisklassen

Agenda

01 Motivation

02 UML Refresh

03 **Modellierung & Produktdefinition mit Faktor-IPS**

Projekt anlegen

Modellierung der Vertragsseite

Modellierung der Produktseite

Tabellen

Aufzählungen

Verwendungen von Formeln

Plausibilisierungen

Vererbung

Testunterstützung

04 Customizing & Tools

Testen mit Faktor-IPS

Unit Testing ohne produktive Produktdaten mit dem InMemoryRuntimeRepository

- Verwendung von JUnit für Modultests / einzelne Funktionen
- Testfälle können unabhängig von produktiven Produktdaten sein
- Isoliertes Testen möglich, Testfälle verwenden nicht die gleichen Testdaten
- Testen von Spezialfällen möglich (die möglicherweise durch produktive Produktdaten (noch) nicht abgebildet werden)

Fachliche Tests mit dem Faktor-IPS Testwerkzeug

- Integrationstest von Produktdaten & Modell/Sourcecode
- Fachliche Tests können vom Fachbereich erstellt werden
- Testfälle und die benötigten Testdaten bilden eine Einheit und stehen unter gemeinsamer Versionskontrolle

Testen mit dem InMemoryRepository

- `// Repository erzeugen`
- `InMemoryRuntimeRepository repository = new InMemoryRuntimeRepository();`
- `// Produkt erzeugen`
- `HausratProdukt produkt = new HausratProdukt(repository, "4711", "HR-Optimal", "2023-01");`
- `// Produkt im Repository ablegen`
- `repository.putProductCmpt(produkt);`

Demo: Unit Tests ohne Produktdaten

Test der Method `getVorschlagVersicherungssumme()`

- Neuen Junit-Testfall anlegen
- `InMemoryRuntimeRepository` anlagen und befüllen

Demo: Fachliche Tests mit dem Faktor-IPS Testwerkzeug

- **Im Modellprojekt einen Testfalltyp *BerechnungsTest* erstellen**
 - Eingabe und Erwartete Attribute anlegen
- **Testfalltyp Klasse implementieren**
 - Methode: executeBusinessLogic()
 - Methode: executeAsserts()
 - Zugriff auf zusätzlich angelegte Attribute für direkt berechnete Werte
 - Notation für Zuordnung Wert ↔ Eingabefeld
 - Methode: setRepository überschreiben
- **Im Produktdatenprojekt Testfall erstellen**

Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

04 Customizing & Tools

Customizing der Produktdefinitionsperspektive

Kopieren von Produkten

Suchoptionen und Views

Erweiterte Codegeneration Optionen

Customizing der Produktdefinitionsperspektive

■ Kategorien

- Mit Kategorien können *Attribute*, *Wertebereiche*, *Berechnungsvorschriften*, *Tabellenreferenzen*, *Regeln*, die einen fachlichen Zusammenhang haben, auf dem Produktbausteineditor in einer Sektion mit einer Überschrift dargestellt werden.
- Kategorien gibt es pro Produktbaustein-Typ. Sie können im Produktbaustein-Typ Editor unter dem Reiter „Kategorien“ bearbeitet werden.

■ Icons

- Für jeden Produktbaustein-Typ kann ein spezielles Icon im Produktbaustein-Typ Editor angegeben werden. Das Icon erscheint in allen Faktor-IPS-Ansichten in denen der Produktbaustein-Typ dargestellt wird.

Customizing der Produktdefinitionsperspektive

■ Labels

- Für viele Modellelemente z.B. Attribute, Assoziationen, Modelltypen, ... können Labels hinterlegt werden. Labels werden anstatt der Modellnamen in den Ansichten und Editoren der Produktdefinitionsperspektive angezeigt.
- Labels haben 2 Aufgaben
 - Fachlich ansprechende Beschreibung z.B. anstatt „versSumme“ wird „Versicherungssumme“ angezeigt
 - Unterstützung von Internationalisierung. Labels können in beliebigen Sprachen angelegt werden und entsprechend der Landeseinstellung wird das Label in den Ansichten und Editoren angezeigt.

Demo zu Customizing der Produktdefinitionsperspektive

- **Kategorien**

- Anlegen der Kategorie Beitragsberechnung in den Deckungstypen
- Verschieben der Tabellen und Formeln in diese Kategorie

- **Icons**

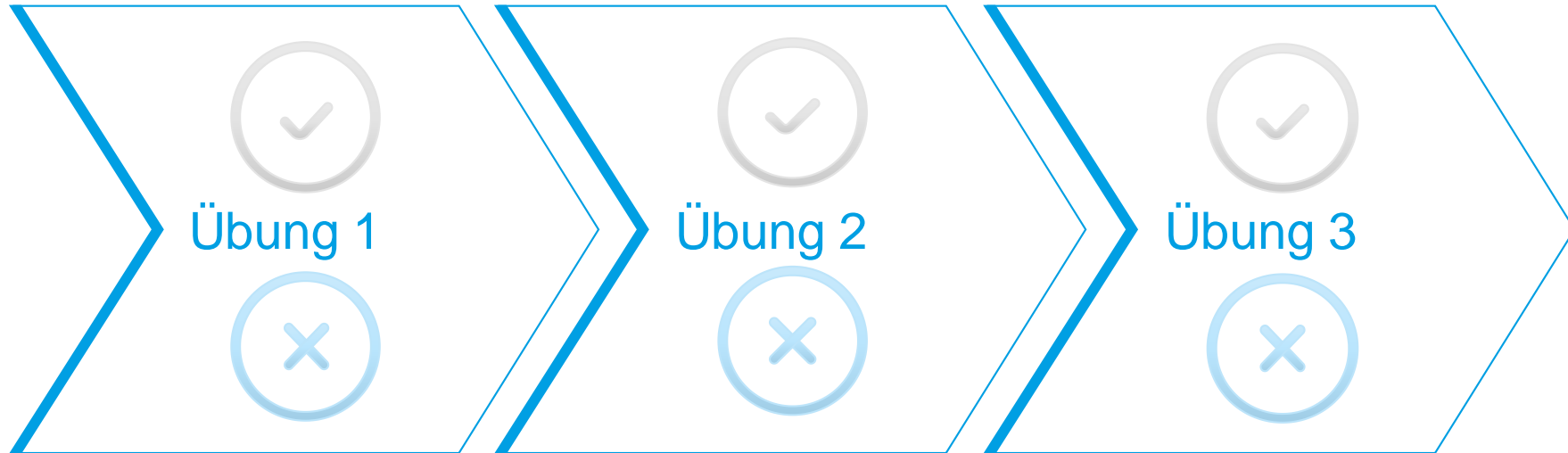
- Anlegen von Icons für die Produktbaustein-Typen

- **Labels**

- Anlegen von Labels für Versicherungssumme, Maximale Versicherungssumme und den Versicherungssummenfaktor

Übungen zu Kapitel IV.A

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01

Motivation

02

UML Refresh

03

Modellierung & Produktdefinition mit Faktor-IPS

04

Customizing & Tools

Customizing der Produktdefinitionsperspektive

Kopieren von Produkten

Suchoptionen und Views

Erweiterte Codegeneration Optionen

Kopieren von Produkten

- Typischerweise sind neue Generationen eines Versicherungsproduktes ähnlich der bisherigen Generation.
- Die Vorgehensweise, um eine neue Generation eines Versicherungsproduktes zu erzeugen, ist das bestehende zu kopieren und entsprechende Änderungen vorzunehmen.
- Eine neuen Generation kann neue Produktbausteine enthalten. Z.B. neue Deckungsarten, Zuschläge oder Nachlässe. Diese wiederum können ähnlich bereits bestehender Produktbausteine sein.
- Unterstützung in Faktor-IPS
 - Wizard „Neue Generation erzeugen“: Kopieren von Produktbausteinen mit Angabe einer neuen Generationsversion
 - Wizard „Produkt kopieren“: Kopieren von Produktbausteinen mit einem Suchen/Ersetzen Muster

Demo: Kopieren von Produkten

- Selektiere das bestehende Produkt HR-Optimal im Produktstrukturexplorer
- Wähle über das Kontextmenü „Neue Generation/Version erzeugen“
- Eine Folgegeneration 2024-01 anlegen. Alle Bausteine übernehmen
- Eine neue Zusatzdeckung Wasserbetten durch Kopie der Fahrraddiebstahl-Deckung erzeugen und den Produkten zuordnen.

Übungen zu Kapitel IV.B

Bitte melden Sie sich, wenn Sie fertig sind oder ein Problem haben.



Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

04 **Customizing & Tools**

Customizing der Produktdefinitionsperspektive

Kopieren von Produkten

Suchoptionen und Views

Erweiterte Codegeneration Optionen

Faktor-IPS Tools

- Modellsuche
 - Suche nach Attributen, Methoden, Assoziationen, Regeln, Tabellen im Workspace
- Produktsuche
 - Suche nach Produktbausteinen eines anzugebenden Produktbaustein-Typs
- Instanz Ansicht
 - Zeigt die Produktbausteine eines Produktbaustein-Typs an, der über das Kontextmenü z.B. Modellexplorers ausgewählt wurde
- Hierarchie Ansicht
 - Zeigt die Ableitungshierarchie von Modellklassen an
- Modellstruktur Ansicht
 - Zeigt die Kompositstruktur eines Modells oder Modellausschnitts an

Agenda

01 Motivation

02 UML Refresh

03 Modellierung & Produktdefinition mit Faktor-IPS

04 **Customizing & Tools**

Customizing der Produktdefinitionsperspektive

Kopieren von Produkten

Suchoptionen und Views

Erweiterte Codegeneration Optionen

Codegenerator Optionen (1)

| Option | Beschreibung |
|-----------------|--|
| Change Listener | Support von Listenern, die bei Änderungen am Modell benachrichtigt werden. |
| Copy Support | Erstellt eine Kopie eines Vertragsobjektes inklusive aller Bestandteile. |
| Delta Support | Berechnet die Unterschiede zwischen zwei Vertragsobjekten. |
| Visitor Support | Generiert die notwendigen Methoden für das Visitor-Pattern. |

Codegenerator Optionen (2)

| Option | Beschreibung |
|-------------------|--|
| JPA Support | Datenbankinformationen können im Modell hinterlegt werden und mit die Vertragsobjekte mit JPA in der Datenbank persistiert werden. |
| JAXB Support | Unterstützt die XML Serialisierung mit JAXB. <div>Ab Faktor-IPS 23.6 wahlweise <code>javax</code> oder <code>jakarta</code></div> |
| Formula Compiling | Formeln können entweder direkt in Java-Klassen übersetzt werden oder aus dem XML heraus interpretiert werden. |

Vielen Dank für Ihre Aufmerksamkeit



Daniel Schwering

Product Owner Faktor-IPS

Daniel.Schwering@faktorzehn.de

FAKTORZEHN